CLEI Electronic Journal, Volume 28, Issue 5, paper 2, September 2025

Assessment of Red and Blue Team Training in the Cyber Range Tectonic using Process Mining

Guillermo Guerrero, Gustavo Betarte, Juan Diego Campo
Universidad de la República, Facultad de Ingeniería,
Montevideo, Uruguay, 11300,
{*gguerrero,gustun,jdcampo*}*@fing.edu.uy*

Abstract

Cyber ranges are specialized computer systems designed to simulate realistic cybersecurity scenarios for training purposes. An in-depth evaluation process is essential for determining whether users have met their training objectives. This article examines the implementation of Tectonic, an academic cyber range, along with a thorough assessment methodology for both offensive and defensive scenarios. This approach employs process mining to analyze training activities from various perspectives. The methodology was applied to training sessions conducted within Tectonic.

Keywords: Cyber Range, Tectonic, Cybersecurity training, Evaluation, Process Mining.

1 Introduction

Cyberattacks perpetrated by malicious actors are increasing each year. It is projected that, by 2025, the total global cost of these attacks will reach $10.5 trillion annually [1]. The primary targets of these attacks are critical infrastructures, particularly the healthcare, financial, and telecommunications sectors [2]. Therefore, cybersecurity professionals play a crucial role in countering these threats.

There is a severe shortage of trained cybersecurity professionals worldwide, leading to an estimated 3.5 million unfilled jobs due to this talent gap [3]. Thus, it is essential to train and educate new cybersecurity professionals. In this field, it is imperative to understand how to prevent cyberattacks and how these attacks are executed. Consequently, cybersecurity professionals must receive training and develop expertise in offensive strategies and defensive measures.

The training provided should not be purely theoretical. However, providing practical security training can be challenging as it must be done in a safe and legal environment. To address this need, cyber ranges have emerged as highly effective platforms. These platforms provide users with computing environments that allow them to acquire new knowledge and explore topics already studied in a safe, legal, and hands-on way.

Numerous cyber range platforms exist, each with distinct characteristics and a specific target audience. The Computer Security Group (GSI) of the School of Engineering of the University of the Republic (FING) has designed and implemented its cyber range platform: Tectonic [4, 5]. This platform has been implemented with state-of-the-art virtualization and cloud technologies. Tectonic complements the theoretical knowledge that university students acquire in various undergraduate and postgraduate cybersecurity courses, offering comprehensive training in offensive and defensive techniques.

We have developed a methodology for assessing cyber range training that integrates process mining with the MITRE ATT&CK [6] and MITRE D3FEND [7] frameworks. This approach enables us to evaluate users' progress in different cyber range scenarios and determine whether they have achieved the necessary training objectives. The discipline of process mining facilitates the discovery, analysis, and examination of processes, interpreting a process as a sequence of steps undertaken to achieve a specific goal. The MITRE frameworks provide a structured way to categorize both attack and defense actions, based on tactics, overarching objectives, and techniques, specific actions employed to accomplish these goals.

Our methodology facilitates the creation of visual models that depict the attack or defense processes undertaken by students during their training. These processes are derived from the activities performed by the participants, which are articulated using tactics and techniques from the MITRE ATT&CK and D3FEND frameworks. Consequently, it becomes possible to compare the actions of the participants against

reference models that represent the desired activities for achieving the training objectives, allowing for the identification of any deviations. This approach enables instructors to evaluate training in offensive and defensive contexts, emphasizing whether students achieve the training objectives and, more importantly, how they accomplish these goals.

Our proposed methodology introduces several enhancements over existing approaches in the field. Firstly, we harness the MITRE ATT&CK and MITRE D3FEND frameworks to classify the activities of training participants according to the tactics and techniques employed by both attackers and defenders. This framework allows us to tap into the extensive knowledge it contains, facilitating a common language that simplifies comprehension for instructors.

Furthermore, our approach permits the analysis of participants' activities at varying levels of granularity. This capability enables us to represent their actions and generate models at the tactical level, reflecting high-level objectives, or at the technical level, which provides a deeper understanding of how those objectives are achieved.

Additionally, by applying this methodology, we can identify reference models that delineate ideal training activities based on the actual performance of participants. This is especially beneficial in complex scenarios where determining optimal training paths manually may prove challenging for instructors. These reference models can also be augmented with insights from instructors, who act as cybersecurity experts and scenario designers.

While this work revolves around cyber range platforms, the proposed methodology can be extrapolated to other domains. The ultimate research objective is to enable the examination, evaluation, and characterization of both manual and automated attack and defense mechanisms developed in computational environments.

This paper builds upon our previous work [8], where we proposed a methodology for evaluating user training within a cyber range, specifically focusing on offensive scenarios, also known as *red team* scenarios. In that paper, we noted the potential for expanding this methodology to encompass defensive or *blue team* scenarios as a direction for future research. This current article addresses that need by presenting a primary contribution: adapting the evaluation methodology for application in offensive and defensive training. Furthermore, we will present real-world applications of the methodology, accompanied by the outcomes achieved.

The rest of this paper is structured as follows: Section 2 introduces cyber ranges platforms, mainly focusing on Tectonic, our academic cyber range. Section 3 presents the methodology for evaluating user training on cyber range platforms for both offensive and defensive scenarios. The application of the technique is explained in Section 4. Section 5 presents a discussion of related work in the area. Finally, Section 6 concludes and describes further work. [1]

2 Cybersecurity Training Platforms

This section introduces cyber range platforms and, in particular, focuses on the academic cyber range Tectonic.

2.1 A primer on Cyber Ranges

Cyber ranges are platforms designed to provide environments for networks, systems, and applications used to train users in cybersecurity. The fundamental purpose is that the user of these platforms can acquire new knowledge or explore topics already studied in a practical, realistic, safe, and legal way [9]. A cyber range is intended for three main groups of people. The first group comprises educators who want to offer their students hands-on practice in computer security to reinforce theoretical knowledge. The second group includes professionals and students who wish to learn, experiment, and improve their skills. The third group includes organizations that must train security teams and assess their abilities [10]. Several actors interact with the platform during its operation: cyber range instructors such as teachers and organizations provide theoretical knowledge to users (students and professionals) through lectures and seminars. To reinforce this knowledge, instructors create training scenarios that present computer security problems related to the topics taught. Users can apply their knowledge and develop skills by engaging in these scenarios. As such, cyber ranges are an effective way to build knowledge and skills in computer security [11]. Notably, a third actor, the cyber range administrator, manages the platform and ensures its smooth operation.

[1] The following changes were made to the article [8]: Sections 2.A and 3.A have been combined into sections 2.1 and 2.2 of the present article. Additionally, Section 2 has been expanded to incorporate further details about the Tectonic cyber range in Section 2.3, along with a comparison to other cyber ranges in Section 2.4. The content from Sections 2.B and 3.B of the previous article is reflected in Section 3 of this document, which now includes new insights into the application of the methodology in defensive scenarios. Section 4 of the earlier article has been enhanced with a detailed description of a case study centered on defensive situations. The other sections have undergone minor modifications.

There are many types of cyber ranges, each with unique qualities and intended for various target audiences. Here is a brief overview of some of these platforms. CyberWiser.eu [12] is a cyber training platform designed for organizations. It provides computer security professionals with training and enables organizations to evaluate their cybersecurity practices. Masaryk University has developed an open-source cyber range called KYPO [13] since 2013. The KYPO platform is based on modern approaches such as containers, infrastructure as code, microservices, and open-source software. Clusus [14] is a cyber range that caters to university students. It enables scenarios of computer security practices to be automatically deployed on cloud services, and results are reported to the Moodle platform when the student completes the scenario. CyTrONE [15] is a framework designed to automate content generation tasks and computer security training environments. Threat Arrest [16] is another cyber range intended for organizations. It prioritizes defensive training, and its methodology involves analyzing an organization's security posture to identify weak points and implementing a customized training program based on that analysis.

2.2 Designing and developing an academic Cyber Range

The GSI research team has operated a cybersecurity laboratory since 2007. This laboratory provides students with practical training in various topics taught in undergraduate and graduate courses to prepare them for real-world scenarios. In 2020, we started the development of a sophisticated, full-fledged cyber range based on state-of-the-art virtualization and cloud technologies to act as the core of the cybersecurity laboratory. The name of our cyber range is Tectonic [5], and it has been made available to the community to support universities and other organizations interested in providing cybersecurity training.

The main functionality of Tectonic is to automate the creation, deployment, and execution of cybersecurity scenarios. Experience has shown us that designing and assembling cybersecurity laboratory scenarios is costly. Most of the cost goes into creating and testing these scenarios, which requires expertise in security, teaching, and advanced skills in configuring and administering networks and operating systems. The result is often difficult to maintain, modify, update, and replicate. To overcome these limitations, finding ways of efficiently managing scenarios throughout their life cycle is necessary. Our solution is to incorporate or develop tools inspired by the infrastructure as code (IaC) approach, which allows for the specification of all the components of a cybersecurity scenario in a declarative manner. This specification is made in a high-level language that can be interpreted and allows for the automatic generation of scenarios on the laboratory underlying platform. A declarative description of the scenarios makes them easily versioned, maintained, and shared, facilitating collaboration with other institutions and laboratories of this type.

The scenarios deployed are realistic and enable experimentation with real security systems and tools. Trainees can carry out sophisticated attacks and exercise varied defense and mitigation mechanisms. However, it is equally important that these environments are entirely secure and that the infrastructure of the organization that hosts them is not adversely affected by the execution of potentially dangerous scenarios and tools. For instance, when studying computer viruses or performing attacks that exploit vulnerabilities in current systems, ensuring that the hosting organization's infrastructure is not compromised is crucial.

In addition, the architecture of the platform is designed in such a way that it efficiently utilizes the available hardware. This approach allows the laboratory to be scaled to support the concurrent execution of multiple scenarios and enable numerous people to participate in training and courses. We also make the platform remotely accessible, allowing for online learning and training. Ideally, it should also manage the allocation of physical resources in a transparent and real-time manner based on demand, but this is not yet ultimately achieved at the time of writing.

Each participating student or group usually requires a dedicated replica or instance of a scenario. These replicas can be personalized with different data to individualize it. For example, if a scenario requires a student to exploit a particular vulnerability to access a specific file, the file's content might differ for each participant. For this, the cyber range supplies mechanisms that enable the definition of personalized content and automates its deployment in the laboratory infrastructure.

The cyber range also offers comprehensive functionalities that support and enhance the delivery of courses and training. One key feature is publishing the necessary teaching material, allowing instructors to easily upload and manage content such as lecture notes, presentations, videos, and quizzes. The platform also provides a content management system and tools for monitoring and evaluating students' performance. This includes progress tracking, grade tracking, and assessment tools that enable instructors to create and administer quizzes, tests, and assignments.

In addition, the platform contains tools that allow for the automatic execution of offensive and defensive activities, creating realistic scenarios where students must react in real time to these automatic processes.

2.3 Tectonic's architecture

Tectonic is a comprehensive platform that facilitates the management of the entire lifecycle of training scenarios, from creation to deployment. It also offers features for monitoring and evaluating student performance and conducting automated attacks on the training infrastructure. Overall, the platform is designed to foster a collaborative and interactive learning environment, allowing instructors to actively engage with their students and deliver high-quality courses focused on both offensive and defensive training.

Figure 1 illustrates the cyber range solution's various elements. The technologies used in the implementation and the different use cases carried out by student users and instructors are also depicted. The cyber range components are categorized into five layers, each fulfilling a vital function in the platform's operation. The following is a concise overview of each layer. For further information regarding the components and functionalities of our cyber range, please consult [4].

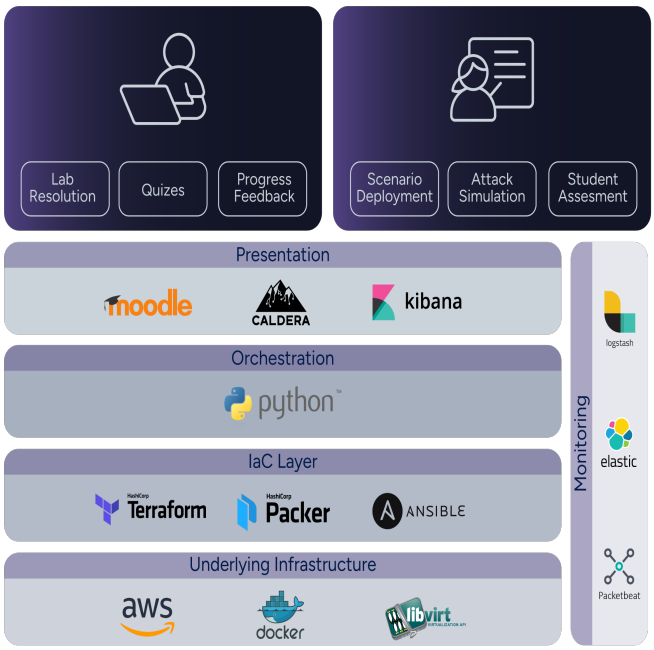


Figure 1: Tectonic components

The underlying infrastructure constitutes the final infrastructure on which the systems and networks that form the basis of a particular scenario are deployed. Users can deploy these systems and networks in the AWS [17] public cloud or on-premises using Libvirt [18] or Docker [19]. Future support for other clouds, such as Azure [20], is planned. Tools such as Packer [21], Terraform [22], and Ansible [23] are used to facilitate automated deployment of the infrastructure, enabling users to manage the systems and networks to be deployed, as well as the configurations to be applied to them. A Python [24] component orchestrates these tools and manages the life cycle of the scenarios, including their deployment, elimination, powering on, powering off, and listing information. The scenarios themselves are described by a specification that allows users to declare various aspects, such as the machines to be deployed, the networks used to connect them, and the configurations to be applied to the machines, among others.

One of Tectonic's key features is its real-time monitoring and evaluation tools for student training activities [25]. To facilitate this, we utilize the Elastic Security tool based on Elastic Stack technology [26], which functions as a Security Information and Event Management (SIEM) system. It can collect events from various sources, normalize them, and correlate them using security analytics rules to detect malicious activity [27].

Event collection can be achieved through two primary methods: utilizing Packetbeat [28] or employing Elastic Agents [29]. The first method, Packetbeat, focuses solely on collecting network traces, making it a more unobtrusive option since students remain unaware of being monitored. In contrast, using Elastic Agents provides greater visibility into student activities, enabling the collection of various events, logs, and metrics. These agents allow for the collection of events associated with executing processes, establishing network connections, creating and deleting files, and more. Additionally, they can collect logs and metrics generated by different services, such as HTTP access logs from web servers. However, this method requires the installation of the agent on the monitored machines, which may interfere with the scenario and is more

resource intensive. All the collected information is then sent to the SIEM, where security rules or analytics correlate the data and generate alerts to detect specific behavior. Instructors can analyze these alerts to gain insights into students' activities and provide feedback on their performance.

The presentation layer of the cyber range consists of various components that act as a frontend interface for educators and students. Among these components is Moodle [30], a platform used to provide theoretical material associated with the scenario and conduct questionnaires for students. Moodle is also used to receive the delivery of flags, information found in one of the systems that make up the scenario and must be delivered by students to complete the training. Another component available in the current version of the cyber range is Kibana, which shows all the data and alerts stored and generated by Elastic Security. We also incorporate Caldera, *an adversary emulation platform designed to run autonomous breach-and-attack simulation exercises efficiently* [31]. This is a significant feature of Tectonic, the execution of automated attacks on the scenario instances, allowing for blue team-type scenarios where the student takes the role of the defender [32]. At the moment, secure Shell (SSH) or Remote Desktop (RDP) clients are required to access the scenarios. In the future, we plan to add Apache Guacamole technology [33] to allow students to access training instances directly from a web browser.

2.4 Distinctive features of Tectonic

Upon reviewing the characteristics of the various cyber ranges, it is evident that there are significant distinctions between these platforms and ours.

Specifically, CyberWiser.eu and Threat Arrest are tailored to prepare and assess organizations. Our cyber range is primarily designed for undergraduate and graduate students but can also accommodate organizational training. On the other hand, KYPO, Clusus, and CyTrONE are all centered around the same user target group as our platform.

While KYPO lacks attack emulation capabilities, its user-friendly graphical user interface (GUI) may offer a more convenient experience than our command-line interface (CLI). Additionally, KYPO facilitates user training and progress evaluation by analyzing collected training event data, leading to an in-depth understanding of student activities and training objectives. Clusus has limitations in simulating attacks; but it offers robust features for training evaluation. It enables the monitoring of training infrastructure through agents installed on machines, facilitating the tracking of participants' activities effectively.. CyTrONE provides attack emulation through its CyPROM feature; however, its evaluation of user training depends on submitted flags and the questionnaires provided via Moodle.

3 Methodology and techniques for cybersecurity training evaluation

This section describes our methodology for evaluating a user's training within a cyber range. We employ process mining and MITRE frameworks as our core tools. This approach allows instructors to assess whether users have met their training objectives and gain insight into their strategies for achieving them.

3.1 Training Evaluation Methodology

The primary goal of the cyber range is to offer a realistic environment where students can practice specific cybersecurity challenges. The process typically involves creating multiple instances of a scenario, one for each student. However, due to hardware limitations, students are usually grouped into massive courses. Although the scenarios are identical, students must find flags unique to each instance to prevent collusion. These flags are information hidden within one of the systems that make up the scenario, and students must locate them and deliver them to complete the training.

Students solve the scenarios through complex actions, using any technique and tool to meet training objectives. These actions typically include, in addition to the minimum ideal steps needed to obtain the flags, reconnaissance activities, trial-and-error, and dead ends, which are also an essential part of the training experience. User training is a complex process that can be studied and analyzed from various perspectives, making process mining a suitable technique for addressing it.

Process mining encompasses various techniques for discovering, monitoring, and enhancing processes by utilizing data collected in an event log as input [34]. A process is a sequence of activities designed to achieve a specific outcome. This approach can be applied across diverse domains and types, from vehicle manufacturing to patient pathways in healthcare settings. In our case, we use it for user training in the cyber range. Traces of process execution are documented in a record known as the event log. Each entry in the log must include an associated activity (i.e., an action or step carried out in the process), a case (i.e., a specific instance of the process), and a timestamp (i.e., the moment when the activity was performed). Process mining tools are

employed to analyze this data, facilitating a deeper understanding of the underlying processes and identifying opportunities for improvement. There are three primary activities related to process mining:

- Process discovery: This technique creates models representing activities conducted within a given process based on the event log. The generated models may include Petri nets [35], Business Process Model and Notation (BPMN) models [36], process trees [37], or similar representations.
- Conformance checking: This technique assesses whether the events captured in the log accurately reflect a pre-existing model and vice versa. The main objective is to ensure consistency between the model and the events recorded in the log. There are two prominent algorithms utilized for the detection, quantification, and analysis of deviations: token replay [38] and alignments [39].
- Process enhancement: This technique refines and expands an existing process model to align it with any newly captured information in the event log.

Figure 2 illustrates the proposed method that comprises three key activities: monitoring and evidence gathering, discovery and verification by reference. The methodology leverages technologies and tools such as Elastic Security, the MITRE ATT&CK and D3FEND frameworks, and process mining. The proposed methodology significantly improves upon the previous assessment performed in Tectonic, which was largely limited to a binary evaluation. The instructor based their assessment solely on the flags submitted by participants, allowing them to determine whether the correct flag had been provided, thereby establishing whether the training objectives had been met. This new approach not only enables instructors to evaluate the achievement of training objectives but, more importantly, provides valuable insights into how students engage in the training by identifying the specific activities and tools they utilize.

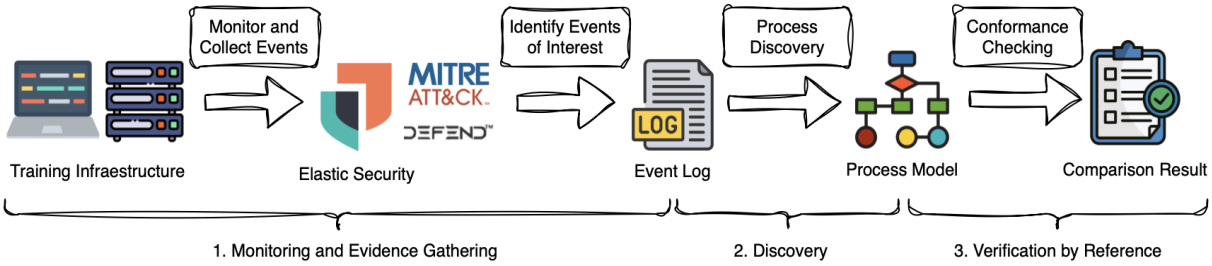


Figure 2: Evaluation methodology

3.1.1 Monitoring and evidence gathering

The initial step in user evaluation is to monitor and collect the events or activities generated during cyber-range training. These events will be used as input for the subsequent tasks involved in this methodology.

The type of information collected depends on the specific scenario or problem being tackled. For example, if a web application is being attacked, the logs generated by HTTP transactions between the client and the web server may be the most helpful information to gather. In other cases, having traces of process executions on a particular machine in the scenario may provide more value for training evaluation.

The collected information is sent to the SIEM, which is stored after normalization. This normalization process is essential as it helps standardize the format of different events, regardless of their nature or technology, making it easier to analyze them later.

We need an event log as input to use process mining techniques. This log should include all the activities performed while training users in the cyber range. However, not all of the events collected are useful for the analysis. For instance, in a scenario where a web application must be attacked, HTTP requests generated by the client that are not attacks might not add value to the analysis. We should only include meaningful events that help understand the activities performed by the user to resolve the scenario. These events could match the ideal resolution activity, which involves obtaining the flags and meeting the training goals. Including only relevant events will prevent the number of events from becoming too large and make subsequent process mining tasks complex and resource-intensive.

Security rules or analytics are run in the SIEM to identify events of interest. These rules are queries that are periodically performed on stored events to identify specific types of activities. The type of activities to identify will depend primarily upon the particular training scenario under consideration. The behaviors

detected through security analytics provide added value, enabling us to contextualize events by understanding their actions, motivations, and associated tasks.

We generate the event log by utilizing the activity identified through the SIEM rules. Each entry in the log comprises three key fields: the timestamp indicating when the event occurred, the activity or action performed by the student, and the case identifier corresponding to the specific instance scenario in which the action was performed. This instance number allows us to identify each student who conducts the training, as each student has a unique training instance scenario. Ultimately, the event log will compile each student's key actions in their training.

3.1.2 Discovery

The next stage of the methodology involves using process discovery techniques and tools to generate models that represent the user training process. This is done by analyzing the event log and applying different process discovery algorithms. For this work, we propose using the Inductive Miner algorithm [40], which identifies the most representative cut in the event log and assigns the corresponding operator. The operator can be sequential, parallel, or a choice between activities. The algorithm continues recursively on each sub-log generated due to the cut made. The result of the application of the algorithm is a model represented by a process tree that ensures the property of soundness. This property implies the correctness of the models based on three essential characteristics: ensuring that the traces finish, there are no deadlocks or infinite loops that prevent a trace from progressing in its execution, and ensuring the absence of non-executable activities [41]. Ultimately, employing this algorithm guarantees the generation of accurate models, which is crucial for the subsequent analyses that can be conducted on them. Models that fail to meet this property can be challenging to interpret and analyze, potentially leading to erroneous results in their assessment.

When it comes to process mining tools, we utilize Disco [42] and ProM Tools [43, 44]. Specifically, we employ Disco for the initial filtering of the event log, while ProM Tools is used for the subsequent stages of process mining. We focus on the Inductive Miner framework [45, 46] within ProM Tools for the analyses. There are multiple process mining tools that can be used to implement the methodology. Our choice to use Disco and ProM Tools is based on several factors: their available features, our prior experience with these tools, and the option to access them for free or through complimentary academic licenses. However, it should be possible to implement this methodology using other process mining tools.

It is possible to use process discovery for two different objectives. First, we can create models that represent users' activities using the event log from a training session. Instructors can then use these models to analyze students' work and evaluate their progress visually. This analysis can focus on various aspects, such as the sequence of tasks executed, the number of attempts made for each task (which can indicate the level of complexity), and the time taken to complete each task. By doing this, instructors can better understand their students' strengths and weaknesses and provide appropriate feedback to improve their performance.

Alternatively, reference models can be generated from event logs of past training sessions. With increased process execution traces, the generated model is expected to represent the underlying process more accurately. These reference models are then used to compare future training sessions. As training sessions progress, the reference model can be improved using process enhancement techniques to incorporate new activities or behaviors that were not previously evident in the model. The instructor who designs and defines the training can manually create reference models. The instructor is responsible for establishing the teaching objectives, activities necessary to meet these objectives, and the computational environment used for training. Therefore, he can use this knowledge to manually define a reference model so that the models capture the ideal activity to meet the training objectives.

3.1.3 Verification by Reference

As the final step in the methodology, we propose verifying the training based on a reference model. This involves taking a model representing the ideal activity required to achieve the different training objectives and an event log capturing the activity carried out by the trainees. We can then apply conformance checking techniques and tools to analyze the process from various perspectives.

One way to carry out the analysis is by focusing on the control flow of the process, which means examining the tasks and their order of execution. This would allow instructors to identify deviations in the training carried out by the users when comparing it with the reference model. For instance, we can identify activities that were not executed, indicating that students failed to complete the training objectives. We can also identify new activities executed or different sequences or orders of execution. This could suggest that students identified new ways of resolving or fulfilling the training objectives that the educator had not considered when generating the practice.

Alternatively, the analysis can be carried out based on the time perspective. This type of analysis could allow instructors to identify users who solve the scenarios too quickly, which could mean that the training is too easy for their level of knowledge. It could also indicate that they potentially cheated and already knew the solution. We could also identify students who are stuck in a particular activity that is very complex for them. Ultimately, the analysis conducted by the instructor through this methodology can serve as input for providing feedback to the students regarding their training.

3.2 Evaluating offensive and defensive scenarios

Up to this point, the methodology has been outlined in a general context without exploring its specific application in various types of cyber range scenarios. There are fundamentally two primary types of scenarios: red team (offensive) and blue team (defensive). Figure 3 illustrates the key aspects of each scenario type, emphasizing high-level objectives, examples of tasks performed by participants, examples of roles exercised and the MITRE frameworks used in the evaluation.

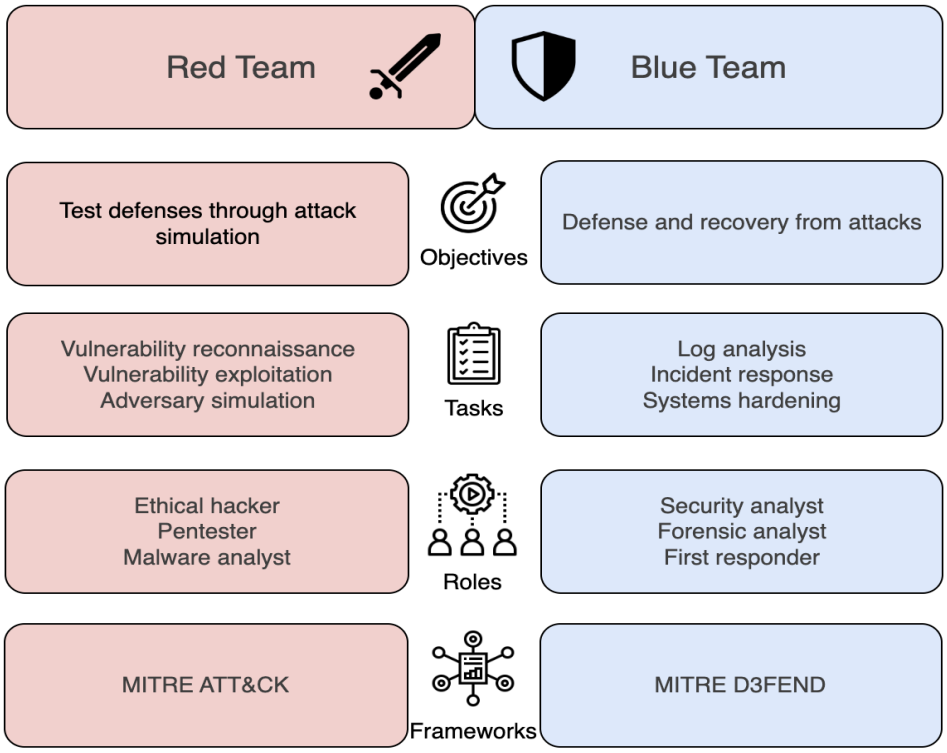


Figure 3: Comparison of scenario types

Identifying events of interest is the crucial aspect that distinguishes the application of our methodology based on the specific scenario. The rules implemented in the SIEM are designed to detect particular types of behaviors or activities, which vary based on the scenario at hand. In a red team scenario, where the student adopts the role of the attacker and engages in offensive operations within the training environment, we aim to identify the attacks they perpetrated. Conversely, in blue team scenarios, the student assumes the role of the defender, so we focus on recognizing the defensive actions undertaken. The rules utilized in the SIEM for offensive scenarios are aligned with the tactics and techniques outlined in the MITRE ATT&CK framework. In contrast, those for defensive scenarios correspond to the MITRE D3FEND framework.

As detailed above, we build the event log once we have identified the events of interest from applying the SIEM rules. Each event in the log has three already detailed fields: a case identifier, an activity, and a timestamp. In our case, the activity field corresponds to the tactics and techniques outlined in the MITRE ATT&CK or MITRE D3FEND frameworks, as determined by the rules configured within the SIEM. When we apply process discovery to generate visual models of the students' training, these models are articulated using these specific tactics and techniques.

The decision to utilize these frameworks stems from their extensive knowledge base, enabling us to represent the activities captured in the event log in a standardized format that enhances analysis and comprehension. Furthermore, applying these frameworks allows us to create models at varying levels of detail. Initially, we can generate models that focus on tactics, which represent the overarching objectives of

an attacker or defender. In cases where a more intricate examination of student training is warranted, an instructor can delve deeper into the activities within the event log by utilizing the techniques specified in the frameworks. These techniques detail the specific actions of attackers or defenders and provide valuable insights into the methodologies students employ to achieve their training goals.

3.2.1 Offensive scenario

MITRE ATT&CK is a comprehensive, globally accessible knowledge base that details adversary tactics and techniques drawn from real-world observations. It is a foundational resource for developing specific threat models and methodologies within the private sector, government agencies, and the cybersecurity products and services community [6]. MITRE ATT&CK framework characterizes malicious activity through Tactics, Techniques, and Procedures (TTP) adopted by adversaries to compromise a computational environment. Tactics refer to the objectives or goals that attackers aim to accomplish, while techniques and procedures describe how they achieve them. This framework delineates 14 tactics and encompasses 203 techniques. Furthermore, it catalogs malicious cyberattack actors and the tools employed in these attacks.

In red team scenarios, our objective is to identify malicious activities performed by the student on the training infrastructure. To achieve this, we leverage SIEM functionalities, as one of its primary roles is to alert of such activities. A crucial aspect to consider is that Elastic Security offers a suite of rules meticulously crafted by cybersecurity experts already organized according to the TTPs outlined in the MITRE ATT&CK framework. Additionally, instructors can create rules tailored to the specific scenario requirements they wish to implement. The MITRE Cyber Analytics Repository (CAR) is a reference for generating security rules. It is an analytics repository based on the MITRE ATT&CK framework [47].

3.2.2 Defensive scenario

MITRE D3FEND complements the MITRE ATT&CK framework by focusing on defensive techniques and strategies. It is a comprehensive knowledge base that documents cybersecurity countermeasure techniques. It serves as a catalog of defensive cybersecurity strategies and delineates their interconnections with offensive or adversarial techniques outlined in MITRE ATT&CK. Like its counterpart, the MITRE D3FEND framework outlines the tactics and techniques defenders employ to safeguard infrastructure. Tactics refer to the maneuvers that defenders execute in response to an adversary, essentially "the what" of an action. In contrast, techniques describe the methods used to implement those tactics, essentially "the how" of carrying out the strategy [7].

In blue team scenarios—where students undertake defensive measures on the training infrastructure—the goal is to recognize these key activities. For instance, we aim to detect actions such as configuring a firewall or blocking user accounts on the system. Creating rules is of utmost importance in this context, as Elastic Security does not come with predefined rules for detecting defensive activities. Nonetheless, this does not preclude us from utilizing the SIEM for this purpose, as the defensive actions taken by students involve executing processes, establishing network connections, and creating files, among others. All these activities can be easily correlated and analyzed within the SIEM. The rules that the instructor generates must be manually mapped to the tactics and techniques of the MITRE D3FEND framework.

4 Experimentation

In this section, we will discuss the implementation of the proposed methodology for evaluating cyber range users and present its application in two scenarios: one focused on red team operations and the other on blue team operations.

4.1 Red team scenario

The training session was conducted for the Network of Excellence in Cybersecurity of Latin America and the Caribbean (Ciberlac) in November 2023. Ciberlac [48] is a collaborative platform that brings together different universities and research institutions working in cybersecurity. The main goal of this network is to foster collaboration between various academic and research institutions to enhance knowledge and experiences related to cybersecurity.

Seven teams of undergraduate students from various universities in the region, all focused on careers in cybersecurity, participated in a three-hour training session. The session used a training environment with Tectonic deployed in the AWS cloud. The exercise involved two machines—an attacker and a victim—connected via a private network. At the beginning of the exercise, the trainees had full access to the attacking machine, and their primary objective was to obtain root privileges on the victim machine.

To achieve this goal, students had to execute an attack that consisted of the following key steps:

- Conduct an online brute force attack on the FTP service using the Hydra tool [49]. The objective is to identify a user with access to a backup file that contains password hashes of other users in the system.
- Perform an offline password cracking using the tool John The Ripper [50]. The file obtained in the previous step will be used as input. This will allow the student to get passwords from different users for SSH access to the victim machine.
- Analyze vulnerabilities and insecure configurations on the victim machine. This activity can be carried out using the LinPEAS tool [51].
- Exploit identified vulnerabilities to escalate privileges and gain the root user.
- Obtain the flag in the `/root/flag.txt` file, which has permissions granting read access only to the root user.

Figure 4 represents mapping these key actions with the tactics of the MITRE ATT&CK framework. Since these actions represent key events in achieving the training objectives, they can be used to generate a reference model that describes the ideal activity of a student.

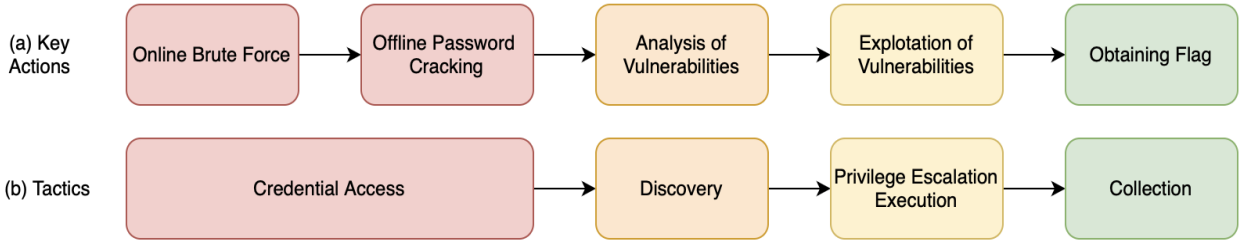


Figure 4: Reference model of red team training

To effectively monitor the activities of students, we installed the Elastic agent on all of the machines they utilized in the cyber range. This tool captures various security events such as command and process execution, network connections, user authentications, and file creation. We then forwarded these events to the SIEM for analysis, employing pre-existing and custom security rules. We used 17 rules tagged with MITRE ATT&CK tactics, including credential access, discovery, privilege escalation, execution, and collection. These rules identify the events of interest that correspond to the key steps of the attack.

The event log was generated solely from the events flagged by the rules operating within the SIEM. The date of occurrence is used as a timestamp, the tactic to which the rule is mapped is used as the activity, and the case identifier corresponds to the victim machine's identifier, which allows for identifying each participating team. The event log for the training session consists of 30,357 events spread across five activities (tactics), as presented in Table 1.

Activity	**Percentage of occurrences**
Credential Access	81.454%
Discovery	18.236%
Privilege Escalation	0.234%
Execution	0.053%
Collection	0.023%

Table 1: Red training - Distribution of events according to activity

By far, the most common actions focused on obtaining user credentials, specifically through brute-force attacks on the victim system's FTP or SSH service. Most groups found this part of the scenario the most complicated and time-consuming. Conversely, the least executed activity is the collection associated with obtaining the final flag. This is because some groups could not acquire the flag, while the task was not particularly challenging for those who did. Once root access to the victim machine is achieved, obtaining the flag is straightforward.

Table 2 outlines the number of events executed per group and Figure 5 the duration of each group's process. Significant variation exists in the number of events and the duration across different groups. For

instance, groups 3 and 7 had fewer events because the SIEM rules failed to detect their brute force attacks. This failure occurred because the rules were designed to identify such attacks only when the number of failed login attempts within a specific timeframe exceeds a predetermined threshold. Since some groups executed several login attempts that did not surpass this threshold, their brute force attacks went undetected. As a result, the start time identified for these groups does not reflect the actual start time of the exercise, and the total process duration is expected to be much lower compared to other groups.

Group	Number of events
1	3,965
2	775
3	35
4	12,505
5	1,165
6	11,858
7	54

Table 2: Red training - Groups number of executed events

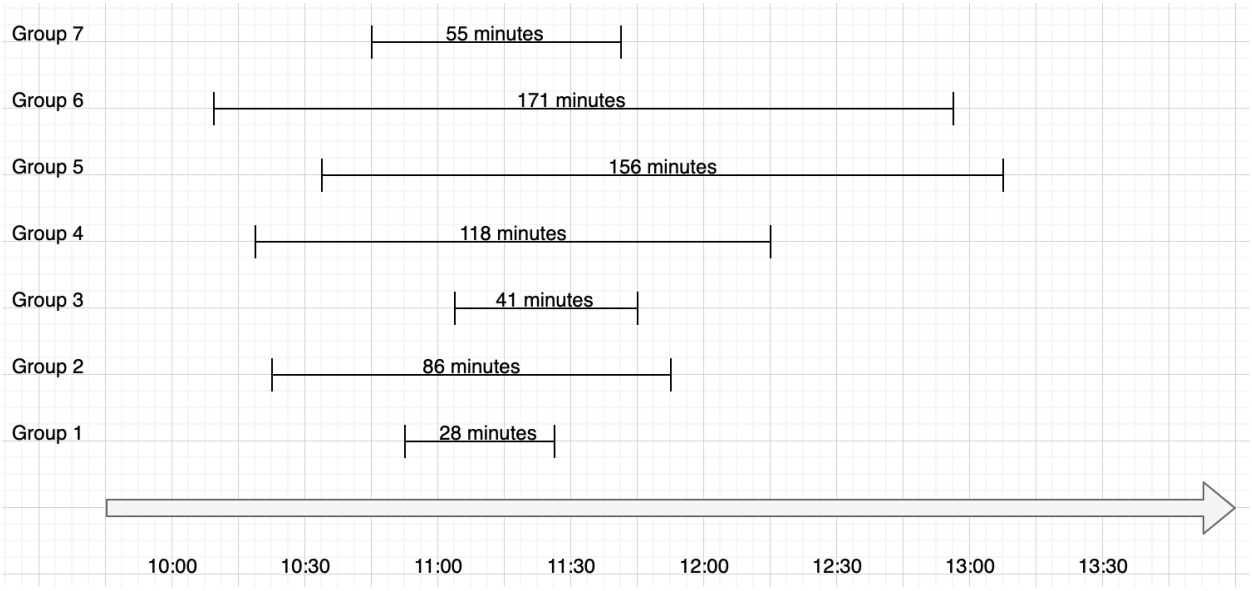


Figure 5: Red training - Timeline

Regarding group 1, which displayed the shortest process duration, it can be inferred that although there is evidence of a brute-force attack and successful access to the victim machine, there are few records of discovery activities on the victim machine and no evidence of tasks related to privilege escalation. This suggests that this process phase may have posed significant challenges for the group, leading to minimal progress. Since there are no records of these activities, their process resulted in the shortest duration compared to other groups.

Finally, groups 4, 5, and 6, which took the longest, displayed a range of behaviors. Group 4 completed the practice satisfactorily, while groups 5 and 6 did not. Specifically, group 6 could not obtain a user from the system, so their brute-force attack was unsuccessful. Their process consisted only of the activity associated with the attempt to obtain credentials.

We employed process discovery techniques to create a model of student activities based on the MITRE ATT&CK tactics. The resulting model is presented in Figure 6, showing the starting point (green dot), end point (red dot), and all the activities that students perform. The numbers associated with activities and paths indicate the number of executions, with darker blue representing the most frequently carried out activities and paths. For additional information regarding the graphical representation and notation employed by the models in this work, please consult [52].

According to the model, the most frequently executed activity in the process is credential access, typically

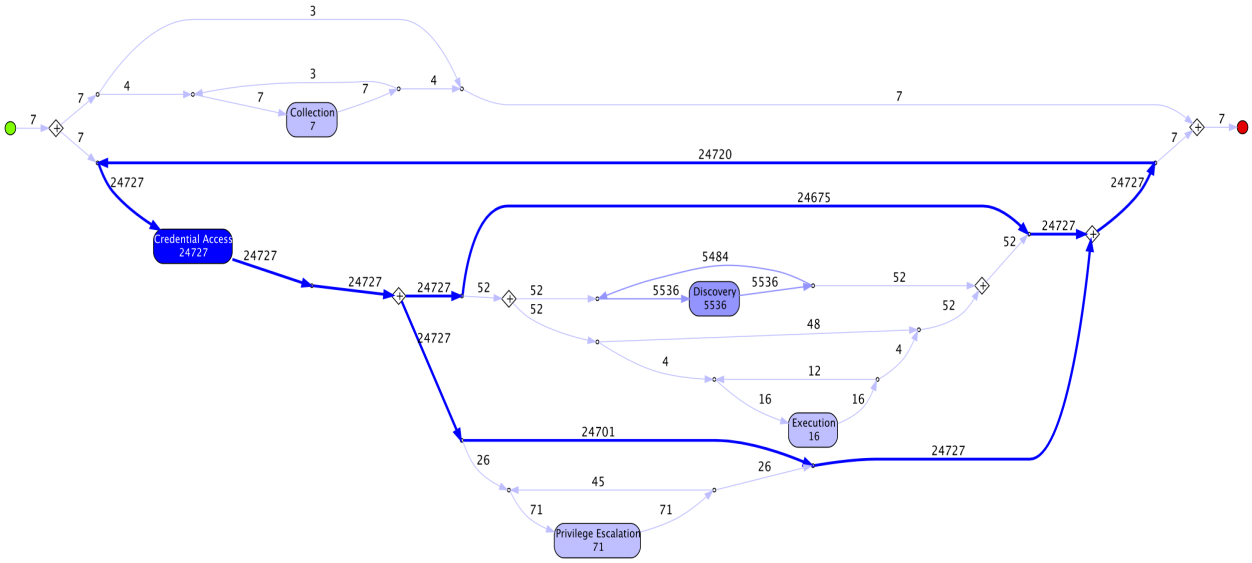


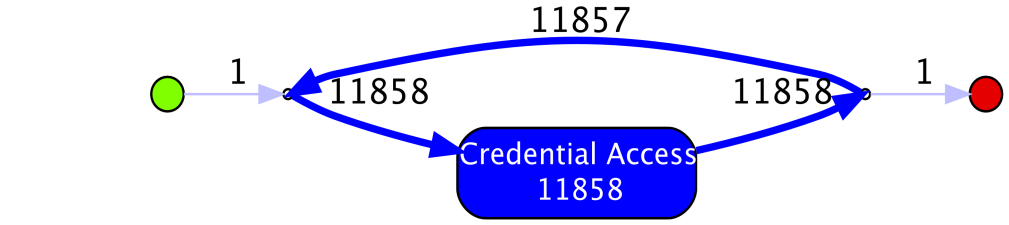
Figure 6: Red training - Students training process model

conducted in a loop. This is due to the nature of brute force attacks, which require multiple attempts with different passwords until a valid one is identified. The second most commonly executed activity is discovery, which involves obtaining information on the victim machine to be used later for the privilege escalation attack. To carry out this activity, the groups used the automatic tool LinPEAS, which performs numerous associated tasks in a single execution. The collection activity involving obtaining the flag was the least frequently executed action. Only four groups executed this action, and only three completed the practice, indicating a false positive in one of the alerted cases and the need for improvements in the rule implementation. The model analysis reveals that the discovery, privilege escalation, and execution tasks occur after at least one credential access event. This aligns with the expected behavior because, in the first stage of the scenario, the student must execute the brute force attack to gain access to the victim machine. Subsequently, they perform the discovery, execution, and privilege escalation tasks to obtain the root user.

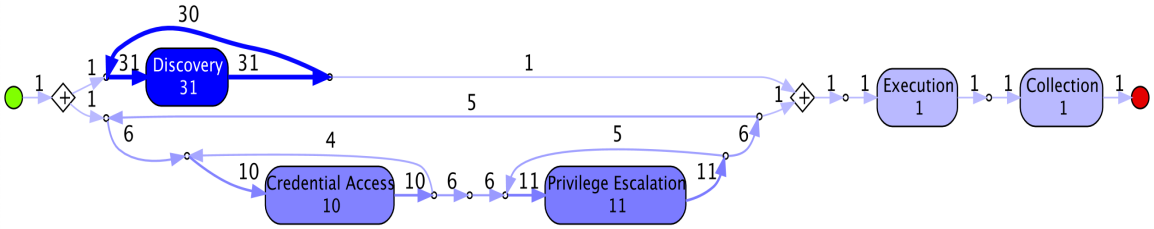
Process models have been created for each group, allowing for a comprehensive analysis of their activities. Figure 7 illustrates the case of groups 6 and 7. These models exhibit entirely distinct behaviors from one another. Group 6 struggled and failed to advance beyond the brute force attack, with only records of activity related to obtaining credentials. In contrast, group 7 recorded various successful attacks, except for brute force, that led to the obtainment of the flag and the correct completion of the practice. Upon examining the models, we found that groups 1 and 5 carried out brute force attacks and could access a user account on the victim machine. However, despite performing discovery tasks on the victim machine, they failed to obtain the root user. Group 3 exhibited similar behavior, minus the brute force attack. Conversely, groups 2 and 4 shared many similarities as they solved the practice successfully. Like group 7, there were records of various activities executed that led to the accurate resolution of the practice.

In this case, the generated training models were not subjected to conformance checking, as no pre-existing reference model was available from past training sessions. However, a manual comparison was made with the reference model shown in Figure 4. This analysis concentrates on the model's control flow, specifically the activities performed and their corresponding sequence of execution, as this is the sole information available. Access to additional reference data, such as execution times, was not available. This reference model was manually created, outlining the tactics employed and their respective execution sequence. Although this reference model may not accurately represent the students' repeated attempts to execute the activities, comparing the reference model and the models of the students who completed the training shows consistency in the activity sequence.

It is worth noting that all the findings derived from the analysis are backed by a debriefing session conducted with the training participants. In this session, several aspects were discussed, such as the groups' achievement of the training objectives, the activities carried out during the training, and the ones perceived to be more challenging and requiring more time.



(a) Group 6



(b) Group 7

Figure 7: Process model of group 6 (a) and group 7 (b)

4.2 Blue team scenario

This training session was conducted during the III Latin American School of Informatics (ELI), which took place in August 2024 in conjunction with the 50th Latin American Conference on Informatics (L CLEI 2024). The school featured a course entitled "Digital Forensics and Incident Response: A Practical Approach". A significant course component was a practical laboratory in which students engaged with a cybersecurity incident.

This laboratory presented a scenario in which an energy company experienced a cyber attack targeting its critical Supervisory Control and Data Acquisition (SCADA) infrastructure. The scenario is based on an Enisa [53] cybersecurity laboratory. To summarize the proposed scenario, the organization operates an industrial network with a master terminal for the remote management of multiple power generation substations. Additionally, the organization maintains a service network that provides mail, FTP, and web services alongside a monitoring network and a corporate network for employee workstations. The attack involved a malicious actor gaining unauthorized access to the master terminal. Students were required to respond to this incident by applying techniques for detection, analysis, mitigation, and recovery, as learned throughout the course.

The scenario was deployed on AWS using Tectonic. While the proposed infrastructure was extensive, only specific components were deployed due to budgetary constraints. A diagram of the deployed infrastructure is provided in Figure 8. Notably, eight copies of the infrastructure were created to accommodate the eight student groups. Students accessed the infrastructure through a designated student access machine, from which they could connect to the master terminal via the SSH protocol. The attacking machine had compromised the master terminal. Initially, the students were unaware of the attacking machine's existence and were expected to identify it as part of their analysis, followed by enacting appropriate mitigation strategies.

It is important to clarify the nature of the attack executed on the training infrastructure. The Caldera framework was utilized to simulate a predefined attack in real time. The attack followed a series of steps:

- Execution of an online brute force attack against the SSH service of the master terminal.
- Access to the master terminal through SSH using the credentials compromised in the previous step.
- Installation of a backdoor within the cron service, which established a reverse shell connection from the master terminal to the attacking machine every time the service is started.
- Execution of the reverse shell.
- Enumeration of the master terminal and collection of sensitive information was then exfiltrated through the reverse shell.

These activities generated traces and artifacts on the master terminal that students could later analyze.

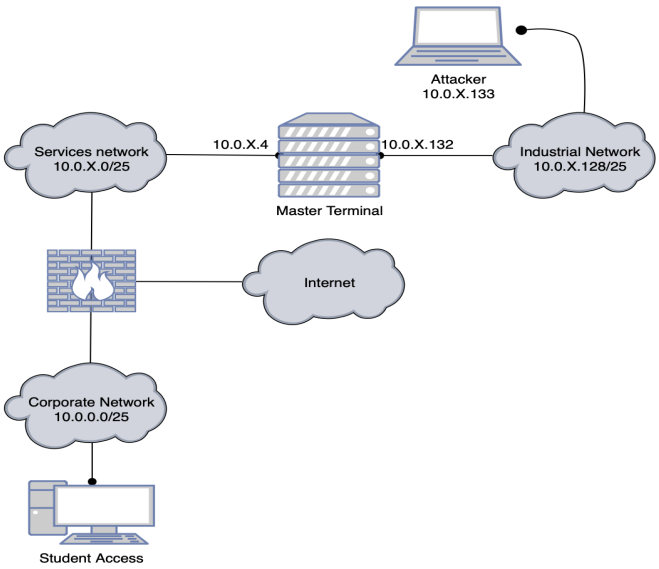


Figure 8: Training infrastructure

The student's objective in this scenario was to effectively respond to the incident by conducting forensics analyses and recovery tasks. Figure 9 illustrates the key activities the students were required to perform, along with their alignment with the tactics of the MITRE D3FEND framework.

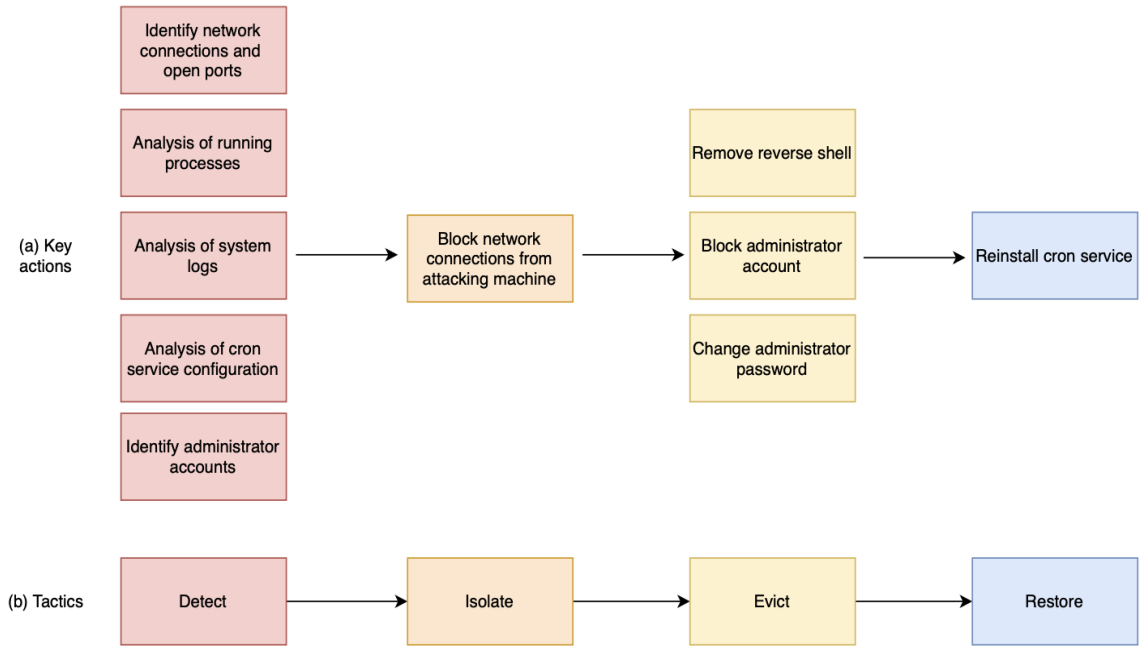


Figure 9: Reference model for blue team training

In detecting and analyzing the attack, the students were expected to examine various components, including running processes, established network connections, system logs, the cron service, and the compromised administrator account. Once they understood the attack, the students implemented mitigation and recovery measures on the master terminal. These measures included blocking incoming and outgoing network connections to the attacking machine, changing the password for the compromised administrator account, removing the reverse shell, and reinstalling the cron service.

Within the Elastic SIEM, we have configured a comprehensive set of sixteen rules meticulously designed to correlate events generated by students interacting with the master terminal. These rules effectively identify the key activities outlined previously. These activities were utilized to construct the event log, including the execution time of each activity, the corresponding tactics and techniques from the MITRE D3FEND

framework, and the instance number linked to the group number that carried out the activity.

An event log comprising a total of 449 recorded events was created. Initially, the log utilized only tactics as activities, which resulted in a mere two activities being represented. However, we recognized that this limited approach would not provide the depth of analysis needed for the instructor's review. Consequently, we opted to enhance the event log by incorporating techniques as activities. As a result, we developed a more comprehensive event log featuring eight activities organized according to the data presented in Table 3.

Activity	**Percentage of occurrences**
Detect - System File Analysis	31.85%
Detect - Connection Attempt Analysis	22.72%
Detect - File Hashing	21.16%
Detect - Process Spawn Analysis	18.04%
Evict - Account Locking	3.12%
Evict - Process Termination	2%
Evict - File Eviction	0.89%
Evict - Credential Revocation	0.22%

Table 3: Blue training - Distribution of events according to activity

It can be observed that only tasks associated with the detect and evict tactics were executed. Compared to the expected activities outlined in Figure 9, the isolate and restore tactics were not implemented. Furthermore, detection activities were executed more significantly, especially those associated with analyzing system logs and configuration files and establishing network connections. This emphasis is due to the instructors' decision to prioritize these activities, as they are more relevant to conducting a forensic analysis—a key topic covered in the theoretical portion of the course. In the final moments of the laboratory session, the instructors informed the students that they could apply mitigation and restoration activities to the infrastructure, which may account for the limited, and in some cases absent, execution of these tactics.

Concerning the group metrics outlined in Table 4 and Figure 10, it is evident that the groups engaged in the execution of between 20 and 104 events, with the time taken ranging from 90 to 166 minutes. Certain groups, such as 1 and 8, undertook very few tasks or required minimal time, concentrating solely on detection activities. In contrast, the majority of the other groups were able to engage in efforts aimed at removing the attacker from the victim machine.

Group	**Number of events**
1	20
2	49
3	104
4	69
5	65
6	67
7	36
8	39

Table 4: Blue training - number of executed events

Process discovery was utilized to develop a model that illustrates user training, depicted in Figure 11. Analyzing this model indicates that the most frequently conducted activities are those related to detection. These include analyzing processes and network connections and reviewing system files such as logs and configuration files. Notably, these detection-related activities occur early in the training process. In contrast, activities linked to the eviction tactic are performed less frequently; only half of the groups engaged in these tasks, and they typically occur towards the end of the process. This pattern is understandable, as these actions involve terminating the process associated with the reverse shell and blocking or changing the password of the compromised administrator account. Students must understand the attack through earlier detection activities to effectively carry out these tasks.

In this instance, conformance checking was not utilized, as no reference model was available for comparison with the student's activities using these techniques. However, a manual comparison was made with the model

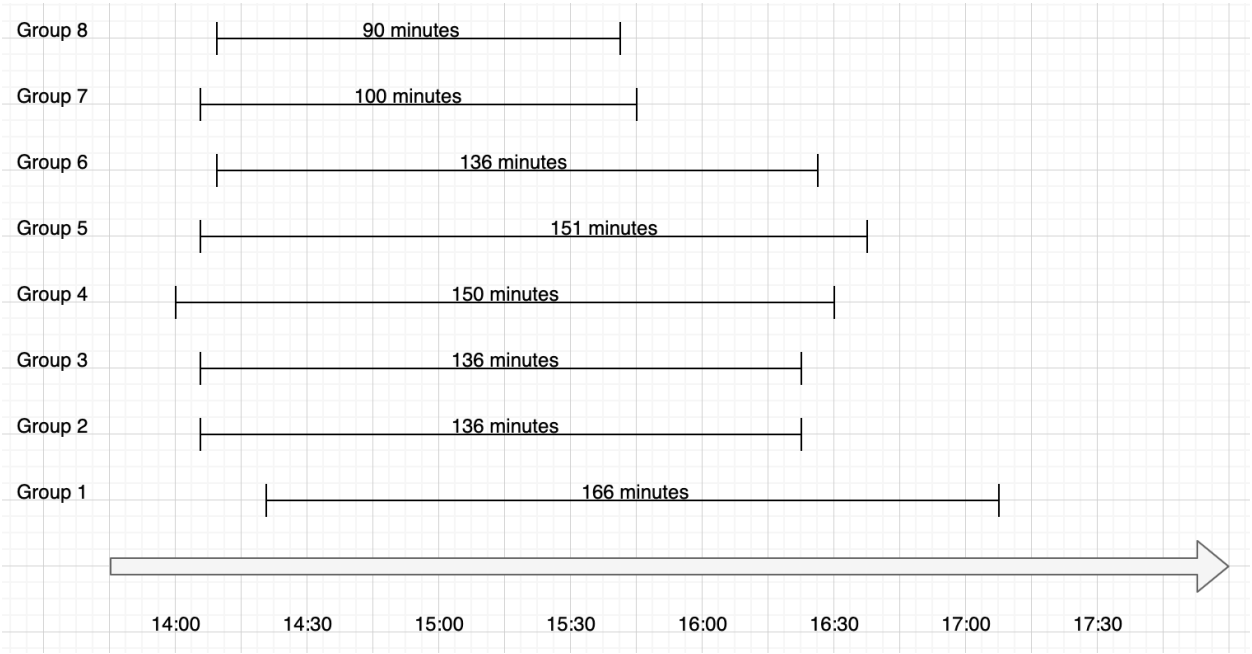


Figure 10: Blue training - Timeline

presented in Figure 9. This analysis focuses on the model's control flow and the sequence of execution, as this is the only information available. There was no access to additional data, such as execution times. This analysis reveals that, although the students did not employ two tactics, the tactics they applied—specifically detection and eviction—were executed in the expected order. It is essential to clarify why file eviction and process termination activities appear "mixed" with detection activities in the model. These activities are linked to the removal of the reverse shell. If the student simply killed the process associated with this reverse shell, it would be automatically relaunched. Therefore, it was expected that the student would make several attempts to eliminate the shell successfully and, after doing so, would take further detection actions, such as listing the processes, to confirm that the removal was effective.

5 Related work

The literature proposes various approaches to evaluating users on cybersecurity training platforms. This section will discuss some of these works and tools for assessing training in cyber ranges.

The paper by Andreolini et al. [54] introduces a methodology for assessing security training using attack models. An attack model encapsulates the malicious actions executed by an attacker, detailing the techniques, tactics, and exploits employed to achieve a specific objective. The authors propose utilizing directed graphs to represent these activities, where nodes signify various states during the training and edges depict the transitions between these states. They develop two types of models: a reference graph and a trainee graph. The reference graph is meticulously constructed by an expert, illustrating the ideal behavior and expected actions that students should demonstrate to fulfill the training objectives. In contrast, the trainee graph is generated automatically by the framework, reflecting the actual activities of the students during the training scenario based on the events recorded. Additionally, the authors present several algorithms designed to compare the reference and trainee models.

In the paper [55], a framework was introduced to detect cheating within a cyber range, incorporating several intriguing concepts for monitoring user activity. The framework utilizes two acyclic directed graphs to delineate scenarios. The first graph captures the network routes of the scenario, while the second represents the attack graph. This attack graph features nodes that denote scenario instances and edges that illustrate the resolution dependencies between these instances. Each instance is linked to a set of flags that users must capture and submit to complete the instance. Typically, these flags are text files in the instance's file system. Each flag is associated with a minimum set of activities that the user must undertake to obtain it and clues that the user can request. These clues provide helpful hints and have a corresponding set of canaries that uniquely identify them, which are utilized to detect potential cheating. Should a user reuse a canary, the framework categorizes it as cheating. Additionally, the framework gathers information regarding the user's

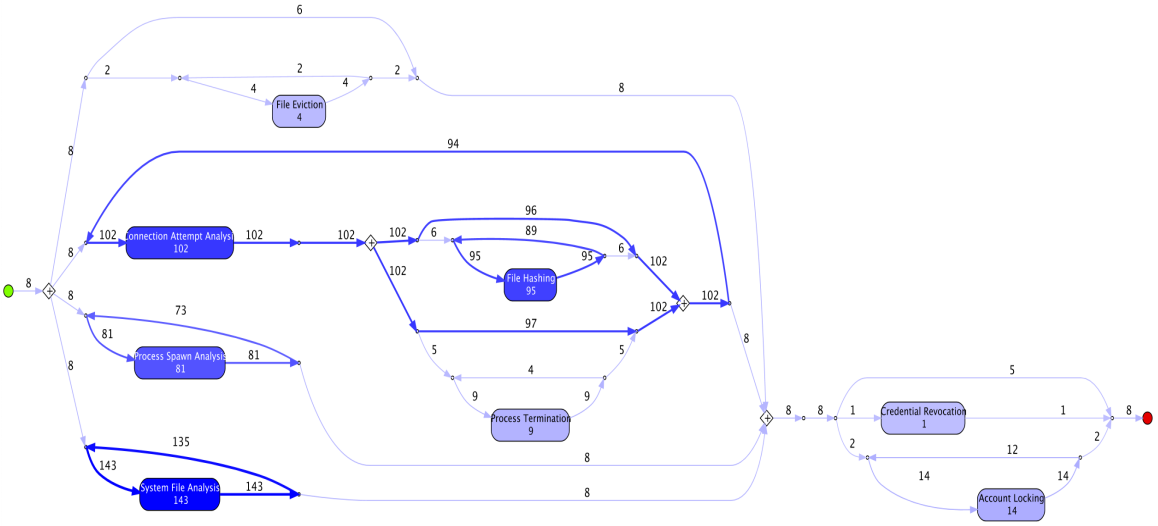


Figure 11: Blue training - Students training process model

activities during the practice session, such as network events, processes, files, and requests for clues. Utilizing this data with the scenario graphs, the framework assesses whether the user has reused canaries or submitted a flag without fulfilling the required minimum activities. In such instances, the framework determines that cheating has occurred.

The publication [56] introduces a method for evaluating students' performance in cybersecurity scenarios that entails gathering the command history of students in a Linux console and constructing directed graphs to illustrate their activities. Instructors can subsequently analyze these graphs to pinpoint deviations from the training objectives.

In [57], the authors propose two methods for evaluating student activity by examining the commands executed in Linux. The first method involves constructing reference graphs that depict the expected activity within a specific scenario. Then, student activity graphs are generated and compared to the reference graph using the history of executed commands. In these graphs, nodes symbolize the sub-goals students must achieve, while the directed edges represent the commands used to accomplish them. The second method involves creating milestone graphs, which include template nodes that indicate the sub-goals within the scenario and the necessary commands to achieve them. These nodes are interconnected with directed edges that signify the order in which the milestones should be completed. Attempt nodes are associated with these template nodes, representing the students' actual activities as they work towards meeting the milestones. These nodes document the number of attempts, the time spent, and the commands executed throughout the process.

Macak et al. [58, 59] conducted a study utilizing process mining to assess student activity within the KYPO cyber range. To accomplish this, the authors propose the creation of event logs categorized into three types of information: events arising from user interactions with the KYPO interface (e.g., flag submissions), executed Linux commands, and commands specifically from the Metasploit tool. Each log entry comprises several fields, including the event execution time, the event category (one of the three mentioned), the specific user activity (a sub-category within the event category), the trainee ID (which identifies the user who performed the event), and any optional event parameters. In this framework, the event field signifies the activity, the timestamp indicates the event's timing, and the trainee ID serves as the case identifier. The authors then employ process discovery techniques to generate models that represent the activities of the trainees in the scenarios.

It is important to note that the studies mentioned above share a common objective of gathering student training data to create models for analysis, often employing directed graphs and sometimes comparing them to reference models. However, significant differences exist between those studies and the one presented in this article.

Firstly, our methodology aims to identify significant events students undertake during their training, utilizing correlation security analytics to detect specific activities. Additionally, these rules are mapped to the MITRE ATT&CK or MITRE D3FEND framework, which security experts designed and implemented. The proposed methodology aims to leverage this knowledge to detect and analyze events of interest generated by students during their training. Instructors can configure these security analytics for each scenario, choosing from predefined rules or creating new ones as needed. Furthermore, by utilizing these frameworks, we can

create models with varying levels of granularity. This allows us first to develop models focused on the frameworks' tactics, which represent high-level objectives, and subsequently produce more detailed models that incorporate the techniques, illustrating how those objectives are accomplished.

Some of the works presented in the study focus on using bash commands executed on Linux systems. However, this approach may have limitations as it may not cover all types of behavior that need to be studied. Examining the executed commands can reveal that a particular script was executed but may not provide insight into the script's activities or the execution result. In contrast, our methodology incorporates a variety of events generated during the training instances, including process executions, network connections, and file generation, among others. This comprehensive approach enables a deeper understanding of student activities.

Some of the studies require that expert instructors build or generate reference models. However, our methodology aims to ensure that these models can be built based on the instructors' knowledge and data obtained from actual student training sessions. This approach facilitates the enhancement of reference models and the inclusion of new insights as they emerge during training. It addresses challenges where instructors struggle to identify specific resolution paths students may discover. We do this by leveraging techniques and tools from process mining to generate reference and student models and study and analyze them. Although only one of the works utilizes this tool, the focus is on the application of process discovery. The other works mainly use ad hoc graph generation and comparison algorithms.

Finally, it is essential to highlight that most existing works concentrate on evaluating offensive training. Our solution facilitates the assessment of both offensive and defensive training, equipping instructors with tools to determine whether students are meeting their training objectives and, more importantly, to gain insights into how these objectives are being achieved. Furthermore, our solution can also help detect instances of cheating among students.

6 Conclusion and further work

The methodology outlined in the previous sections is not just a theoretical construct but a practical and systematic approach that uses advanced process mining techniques and technology, operational observability, and security analytics. By analyzing real-life training sessions, instructors can pinpoint potential gaps and identify areas for improvement, thereby gaining a deeper understanding of the processes involved in conducting these training activities. This systematic approach provides valuable insights and tangible opportunities to take necessary actions to improve the training sessions, making it a crucial tool for cybersecurity academics and professionals.

Moreover, the methodology we have introduced is not a static solution; it serves as a platform for innovation and growth. It can be further developed to support various functionalities beyond its current capabilities. Rather than simply evaluating the performance of a training session, the methodology can be leveraged to dynamically adapt the scenario to incorporate new challenges into the training. This would involve enriching the methodology to combine methods and techniques that make it possible to define and deploy adaptive offensive and defensive scenarios, paving the way for advancements in cybersecurity training.

Several lines of future research are proposed. Firstly, the security analytics in the SIEM for detecting events of interest may produce inaccurate results, leading to false positives or negatives. This affects the models generated, making them less accurate in representing the user training process. We intend to investigate the possibility of leveraging machine learning-based SIEM rules to enhance the precision of detecting malicious events. These rules would complement static rules and learn from the collected data to improve the detection of activity generated by students in their training, thereby addressing the issue of false positives and false negatives.

Furthermore, we propose an avenue for future research to explore analytics that facilitate identifying defensive activities. We have observed that while numerous repositories of security rules are intended for use in a SIEM to detect malicious activities, a similar set of resources appears to be lacking for defensive activities. Establishing repositories containing rules for identifying and characterizing defensive actions would be beneficial.

At present, the methodology is not integrated into Tectonic as an autonomous tool. Instructors are required to manually complete several tasks for its implementation, including exporting Elastic Security alerts to generate the event log, importing these alerts into process mining tools, and conducting event log filtering, process discovery, and conformance checking tasks. This manual process poses a limitation, as it makes the implementation of the methodology more complex. Future efforts will focus on developing a tool that automates these tasks and seamlessly integrates the methodology into Tectonic. This enhancement will enable instructors to access it through a dedicated interface for participant evaluation.

An alternative avenue for future research involves exploring the application of the methodology in more

intricate scenarios. For example, this includes environments where both red and blue teams operate on the same infrastructure, with the red team executing attack actions while the blue team implements defensive measures in response. It would be interesting in this case to construct models that show the interaction between the teams, so as to give trainers visibility over how one team reacts to the actions of the other.

Furthermore, experiments can be conducted in complex situations pertinent to key sectors such as healthcare, banking, and industry, which often rely on critical infrastructure that includes IoT devices. Specifically, within the industrial sector, the methodology could incorporate the MITRE ATT&CK framework's ICS matrix, designed specifically for industrial control systems.

Conversely, another area for enhancement is to offer adaptive training for students. In this approach, training scenarios—whether offensive or defensive—can dynamically change in real time as the training unfolds. This ensures that the scenario evolves following the student's knowledge and skills, maximizing the benefits of the training experience. Techniques from streaming process mining [60] can be used to study the progress of user training in real time. Furthermore, scenarios should be crafted so that, while the final training objective remains singular, various pathways to resolution are available, potentially differing in difficulty. The transformations or actions to be performed in real time to effect changes in the scenario should also be clearly defined. This way, if a student demonstrates a higher level of proficiency than anticipated, a transformation can be implemented to direct them toward a more complex resolution path. For example, in a blue team scenario, an attack may be initiated under specific conditions utilizing Caldera, requiring the student to defend the infrastructure. Conversely, techniques from the moving target defense [61] domain could be employed in a red team scenario, thereby complicating the student's offensive tasks. In short, this enhancement would enable the cyber range to offer adaptive training scenarios that can be dynamically adjusted based on the user's abilities and skills, resulting in improved tailored training.

References

[1] Steve Morgan, "Cybercrime to cost the world $10.5 trillion annually by 2025," https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/, 2020, [Online, last accessed: 30/06/2025].

[2] European Repository of Cyber Incidents (EuRepoC), "Critical infrastructure tracke," https://cit.eurepoc.eu, 2025, [Online, last accessed: 30/06/2025].

[3] Steve Morgan, "Cybersecurity jobs report: 3.5 million unfilled positions in 2025," https://cybersecurityventures.com/jobs/, 2023, [Online, last accessed: 30/06/2025].

[4] G. Guerrero, G. Betarte, and J. D. Campo, "Tectonic: An academic cyber range," in *2024 IEEE Biennial Congress of Argentina (ARGENCON)*, 2024, pp. 1–8.

[5] GSI, "Tectonic," https://www.fing.edu.uy/inco/proyectos/tectonic, [Online, last accessed: 30/06/2025].

[6] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," 2018.

[7] P. E. Kaloroumakis and M. J. Smith, "Toward a knowledge graph of cybersecurity countermeasures," *The MITRE Corporation*, vol. 11, 2021.

[8] G. Guerrero, G. Betarte, and J. D. Campo, "Process mining-based assessment of cyber range trainings," in *2024 L Latin American Computer Conference (CLEI)*, 2024, pp. 1–10.

[9] NICE, "The cyber range: A guide. guidance document for the use cases, features, and types of cyber ranges in cybersecurity education, certification and training," https://www.nist.gov/system/files/documents/2023/09/29/The%20Cyber%20Range_A%20Guide.pdf, September 2023.

[10] ——, "Cyber ranges," https://www.nist.gov/system/files/documents/2018/02/13/cyber_ranges.pdf, 2018.

[11] L. Topham, K. Kifayat, Y. A Younis, Q. Shi, and B. Askwith, "Cyber security teaching and learning laboratories: A survey," *Information & Security*, vol. 35, pp. 51–80, 01 2016.

[12] "Cyberwiser.eu: Cyber range and capacity building in cibersecurity," https://www.cyberwiser.eu/, [Online, last accessed: 30/06/2025].

[13] "Kypo cyber range platform," https://docs.crp.kypo.muni.cz/, [Online, last accessed: 30/06/2025].

[14] R. Flinterman, A. Smit, E. Hildebrand, and J. Mulder, "Clusus: a cyber range for network attack simulations," Delft, Países Bajos, 2019.

[15] C. R. Organization, D. C. J. A. I. of Science, and Technology, "Cytrone user guide," https://github.com/crond-jaist/cytrone/releases, Tech. Rep., 2021.

[16] G. Hatzivasilis, S. Ioannidis, M. Smyrlis, G. Spanoudakis, F. Frati, C. Braghin, E. Damiani, H. Koshutanski, G. Tsakirakis, T. Hildebrandt, L. Goeke, S. Pape, O. Blinder, M. Vinov, G. Leftheriotis, M. Kunc, F. Oikonomou, G. Magilo, V. Petrarolo, A. Chieti, and R. Bordianu, "The threat-arrest cyber range platform," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2021, pp. 422–427.

[17] AWS, "Cloud computing," https://aws.amazon.com/, [Online, last accessed: 30/06/2025].

[18] Libvirt, "The virtualization api," https://libvirt.org, [Online, last accessed: 30/06/2025].

[19] Docker, "Docker: Accelerated container application development," https://www.docker.com, [Online, last accessed: 30/06/2025].

[20] Microsoft, "Azure," https://azure.microsoft.com, [Online, last accessed: 30/06/2025].

[21] HasiCorp, "Packer," https://www.packer.io/, [Online, last accessed: 30/06/2025].

[22] ——, "Terraform," https://www.terraform.io/, [Online, last accessed: 30/06/2025].

[23] RedHat, "Ansible," https://www.ansible.com/, [Online, last accessed: 30/06/2025].

[24] Python.org, "Welcom to python.org," https://www.python.org, [Online, last accessed: 30/06/2025].

[25] R. Gallardo and G. Guerrero, "Reingeniería del laboratorio de seguridad informática análisis, diseño e implementación de un cyber range," Montevideo, Uruguay, 2021, available at https://hdl.handle.net/20.500.12008/30925.

[26] Elastic, "Elastic stack," https://www.elastic.co/elastic-stack, [Online, last accessed: 30/06/2025].

[27] ——, "Modernize secops with elastic security," https://www.elastic.co/security, [Online, last accessed: 30/06/2025].

[28] ——, "Packetbeat," https://www.elastic.co/beats/packetbeat, [Online, last accessed: 30/06/2025].

[29] ——, "Elastic agent," https://www.elastic.co/elastic-agent, [Online, last accessed: 30/06/2025].

[30] Moodle, "Moodle documentation," https://docs.moodle.org/403/en, [Online, last accessed: 30/06/2025].

[31] Mitre, "Caldera," https://caldera.mitre.org, [Online, last accessed: 30/06/2025].

[32] G. Corujo and M. Rodriguez, "Cyra.uy: Hacia un cyber range académico," Montevideo, Uruguay, 2023, https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/42955.

[33] Apache, "Apache guacamole," https://guacamole.apache.org, [Online, last accessed: 30/06/2025].

[34] W. van der Aalst, *Process Mining Data Science in Action.* Springer Berlin, Heidelberg, 2016.

[35] Jörg Desel, Wolfgang Reisig and Grzegorz Rozenberg, *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, 2004.

[36] Remco M. Dijkman, Marlon Dumas and Chun Ouyang, "Semantics and analysis of business process models in bpmn," *Information and Software Technology*, vol. 50, no. 12, pp. 1281–1294, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584908000323

[37] Wil van der Aalst, Joos Buijs and Boudewijn van Dongen, "Towards improving the representational bias of process mining," in *Data-Driven Process Discovery and Analysis*, K. Aberer, E. Damiani, and T. Dillon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 39–54.

[38] A. Rozinat and W. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030643790700049X

[39] Wil Aalst, Arya Adriansyah and Boudewijn Dongen, "Replaying history on process models for conformance checking and performance analysis," *WIREs Data Mining and Knowledge Discovery*, vol. 2, pp. 182–192, 03 2012.

[40] Sander J. J. Leemans, Dirk Fahland and Wil van der Aalst, "Discovering block-structured process models from event logs - a constructive approach," in *Application and Theory of Petri Nets and Concurrency*, J.-M. Colom and J. Desel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 311–329.

[41] S. J. Leemans, "Robust process mining with guarantees," in *International Conference on Business Process Management*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:52196338

[42] Fluxicon, "Disco," https://fluxicon.com/disco/, [Online, last accessed: 30/06/2025].

[43] B. F. van Dongen, A. K. A. de Medeiros, H. M. W. Verbeek, A. J. M. M. Weijters, and W. M. P. van der Aalst, "The prom framework: A new era in process mining tool support," in *Applications and Theory of Petri Nets 2005*, G. Ciardo and P. Darondeau, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 444–454.

[44] ProM, "Prom tools," https://promtools.org, [Online, last accessed: 30/06/2025].

[45] S. J. Leemans, *Inductive visual Miner Manual*, 2017.

[46] Leemans, "Visual miner," https://leemans.ch/visualminer/home/, [Online, last accessed: 30/06/2025].

[47] MITRE, "Mitre cyber analytics repository," https://car.mitre.org, [Online, last accessed: 30/06/2025].

[48] R. CiberLaC, "Red ciberlac," https://ciberlac.org, [Online, last accessed: 30/06/2025].

[49] "Hydra," https://www.kali.org/tools/hydra/, [Online, last accessed: 30/06/2025].

[50] openwall, "John the ripper password cracker," https://www.openwall.com/john/, [Online, last accessed: 30/06/2025].

[51] carlospolop, "Linpeas - linux privilege escalation awesome script," https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS, [Online, last accessed: 30/06/2025].

[52] S. J. Leemans, *Inductive visual Miner & Directly Follows visual Miner*, 2019.

[53] Enisa, "European union agency for cybersecurity," https://www.enisa.europa.eu, [Online, last accessed: 30/06/2025].

[54] M. Andreolini, V. Colacino, M. Colajanni, and M. Marchetti, "A framework for the evaluation of trainee performance in cyber range exercises," *Mobile Networks and Applications*, vol. 25, 02 2020.

[55] N. Kakouros, "A cheat detection system for an educational pentesting cyber range: an intrusion deficit approach," Master thesis, KTH Royal Institute of Technology School of Electrical Engineering and Computer Science, 2020, available at https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-284254.

[56] R. Weiss, M. E. Locasto, and J. Mache, "A reflective approach to assessing student performance in cybersecurity exercises," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 597–602. [Online]. Available: https://doi.org/10.1145/2839509.2844646

[57] V. Švábenský and et al., "Evaluating two approaches to assessing student progress in cybersecurity exercises," in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 787–793. [Online]. Available: https://doi.org/10.1145/3478431.3499414

[58] M. Macak, R. Oslejsek, and B. Buhnova, "Applying process discovery to cybersecurity training: An experience report," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022, pp. 394–402.

[59] ——, "Process mining analysis of puzzle-based cybersecurity training," ser. ITiCSE '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 449–455. [Online]. Available: https://doi.org/10.1145/3502718.3524819

[60] A. Burattin, *Streaming Process Mining*. Cham: Springer International Publishing, 2022, pp. 349–372. [Online]. Available: https://doi.org/10.1007/978-3-031-08848-3_11

[61] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proceedings of the First ACM Workshop on Moving Target Defense*, ser. MTD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 31–40. [Online]. Available: https://doi.org/10.1145/2663474.2663479