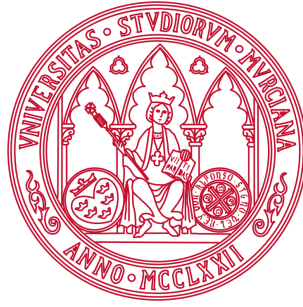




University of Murcia

FACULTY OF COMPUTER SCIENCE

**MIS^{um}. REAL-TIME ANALYSIS AND
MONITORING OF MICROSERVICES
BASED APPLICATIONS**

**A CONCEPTUAL PROPOSAL AND
SCORPION USE CASE**

ACADEMIC YEAR 2024-2025

June exam session

Tutor

José Antonio Ruiperez Valiente

Co-tutor

Félix Jesús García Clemente

Author

Alberto Gálvez Gálvez

Acknowledgements

I would like to sincerely thank my tutors and colleagues for their continuous support, guidance, and companionship over the past two years. Your presence has significantly contributed to my professional and personal growth. It has truly been wonderful getting to know each of you.

My deepest thanks to all the teachers who have inspired in me the desire to learn and understand things more profoundly. Your dedication, patience, and encouragement have been crucial to my journey.

My heartfelt gratitude to my family for always providing unwavering support and guidance, shaping not just my academic achievements but also my character.

Finally, to my friends who have continuously brought joy, laughter, and warmth into my life, thank you for your constant presence and friendship. Your support has enriched this journey in countless ways.

Summary

This dissertation addresses a persistent gap in modern cyber range design: the absence of unified, real-time observability capable of informing both technical administration and pedagogical decision-making. Focusing on microservice-oriented ranges, it proposes a technology-agnostic conceptual framework accompanied by a reference implementation to validate the concept in practice. By integrating Infrastructure-as-Code (IaC) automation, cloud-native monitoring stacks, and adaptive learning analytics, the study positions observability as a first-class concern in cybersecurity education and research.

Three converging trends motivate this investigation. The migration from monolithic to microservice architectures multiplies telemetry sources; the rise of cloud and hybrid deployments enables ephemeral per-student ranges while fragmenting logs and metrics; and workforce development programmes increasingly demand timely insight into learner progress, even though instructors remain largely confined to post-exercise forensic analysis. Against this backdrop, the dissertation argues that an integrated observability pipeline capable of correlating infrastructure traces, adversary actions, and learner interactions can fully realise the pedagogical potential of cyber ranges.

The dissertation surveys the state of the art in cyber range architectures published between 2020 and 2025, identifying recurrent building blocks and shortcomings in current monitoring practices. Building on this review, it formulates a reference architecture that embeds real-time observability as a cross-cutting concern alongside infrastructure, orchestration, user management, and data exploitation. A comprehensive telemetry typology then normalises heterogeneous system-level and user-level data streams into a coherent data model. Finally, the framework is validated on the SCORPION cyber range, a platform that uses Infrastructure-as-Code (IaC) to couple microservice orchestration with gamification and learning analytics.

Within the architecture, a dedicated monitoring tier is situated between the orchestration plane and downstream analytics. The implementation presented in this study employs a stack consisting of Fluent Bit, Loki, and Grafana as illustrative open-source components; however, the underlying blueprint remains compatible with alternative toolchains. Telemetry is funnelled through a Kafka backbone and processed by a Monitor – Analyse – Plan – Execute loop augmented with a shared knowledge base (MAPE-K). This mechanism enables adaptive orchestration while preserving a complete audit trail.

In the SCORPION use case, Terraform orchestrates an Observer virtual machine and per-student machines on a vSphere cluster. Ansible deploys containerised monitoring services on the Observer, while each student machine is provisioned with Bash DEBUG loggers, Ascinema recorders, and lightweight forwarders. SSH sessions are wrapped by a master shell that triggers session capture, thereby enabling stealth monitoring that remains invisible to the trainee. Instructors can consult Grafana dashboards that update in real time with each executed command, as well as the SCORPION instructor UI, which lists every command executed along with its timestamp, host, IP address, and working directory metadata. During pilot exercises, these facilities enabled rapid verification of learner progress, facilitated troubleshooting, and revealed suspicious behaviour indicative of potential cheating, all with negligible performance overhead.

Although the current validation focuses on command-level telemetry and basic gamification metrics, the architecture accommodates richer data streams such as CALDERA adversary logs and biometric signals. Extending connector support to Windows, federating deployments across multiple clouds and mining aggregated traces to derive canonical solution paths constitute planned enhancements.

The study presents a reference architecture grounded in the literature that integrates foundational pillars with emergent modules; a telemetry framework that unifies heterogeneous data through shared identifiers and streaming pipelines; and a demonstration that enterprise-grade observability can be achieved using licence-free components and modest hardware.

Future research will integrate Windows Event Logs and Sysmon, leverage public-cloud providers to host geographically distributed labs, map telemetry to competency frameworks such as NICE or ENISA, and embed lightweight machine-learning models at the edge to deliver context-aware hints and self-healing scenarios.

In conclusion, the dissertation repositions observability from an operational afterthought to a pedagogical catalyst. By demonstrating how real-time data streams—ranging from terminal commands to physiological indicators—can be monitored and utilised for immediate feedback, adaptive orchestration, and rigorous after-action review, it charts a pragmatic path toward cyber ranges that are both insightful and immersive.

Index

List of Figures	12
List of Tables	14
List of Listings	16
Definitions	18
List of Acronyms and Abbreviations	20
1 Introduction	22
1.1 Context and Motivation	22
1.2 Challenges with Dynamic and Real-Time Data	22
1.3 Objectives of the Work	23
1.4 Contributions	23
1.5 Document Organisation	24
2 State of the Art	26
2.1 Definition and Purpose of Cyber Ranges	26
2.2 Classification of Cyber-Range Platforms	26
2.3 Key Features and Emerging Trends	26
2.3.1 Virtualisation and Microservice Platforms	27
2.3.2 Federation and Interoperability	27
2.3.3 Gamification and Adaptive Orchestration	27
2.3.4 Monitoring and Data Exploitation	27
2.4 User Behaviour Monitoring and Learning Analytics	28
2.5 Infrastructure Observability in Microservice-Based Ranges	28
3 Methodology	30
3.1 Scope and Methodological Approach	30
3.2 Identification of Research Works and Initial Screening	30
3.3 Refined Exclusion Criteria	30

3.4	Detailed Rationale for Exclusions	31
3.5	Including an Additional Relevant Paper	31
3.6	Inclusion Criteria and Final Corpus	33
3.7	Review and Coding Process	33
4	Proposal of a Conceptual Cyber Range Architecture	36
4.1	Overview	36
4.2	Summaries of Included Papers	36
4.3	Comparative Tables	41
4.4	Emergent Features and Final Conceptual Architecture	42
4.4.1	Emergent (Inductive) Modules	42
4.4.2	Proposed Extended Reference Model	43
5	Typology for Data Generation, Monitoring, and Exploitation	46
5.1	System-Level Data Representation	46
5.1.1	Provisioning Logs (Terraform)	46
5.1.2	Configuration & Scenario Execution Logs (Ansible)	47
5.1.3	Adversarial Actions (MITRE CALDERA)	48
5.1.4	ICS/OT Telemetry (Industrial Systems)	48
5.1.5	Cloud Platform Events (Multi-Cloud)	49
5.1.6	Federation Metadata (Multi-Range)	49
5.2	User-Level Data Representation	49
5.2.1	xAPI Learning Experience Events	49
5.2.2	HCI Interaction Logs (Mouse, Keyboard, Command Line Interface (CLI))	50
5.2.3	Biometric and Physiological Signals	50
5.3	Monitoring & Analytics	50
5.3.1	Data Normalisation and Correlation	50
5.3.2	Advanced Analytics and Adaptation (MAPE Loops)	51
5.4	Summary of the Data Typology	51
5.5	Conclusion	53

6	SCORPION Use Case	54
6.1	Introduction and Context	54
6.2	Key Technology Decisions	54
6.3	Design	55
6.3.1	Overall Architecture	55
6.3.2	Data Collection Workflow	56
6.4	Implementation	56
6.4.1	Terraform Snippet	56
6.4.2	Ansible Playbook (Observer)	57
6.4.3	Docker Compose (Observer)	58
6.4.4	Ansible Playbook (Student VMs)	58
6.4.5	Shell Wrapper and Command Logger	60
6.4.6	Fluent Bit Configuration	61
6.5	Use-Case Dry-Run	62
7	Conclusions	68
	References	70

List of Figures

1	PRISMA Flow Diagram Adapted for the Systematic Review of Cyber Range Architectures (2020–2025)	33
2	Illustrative Extended Reference Model Integrating the Five Pillars and Emergent Modules	44
3	Training catalogue where the instructor clicks + to deploy an existing exercise.	62
4	Modal for configuring navigation, adaptive mode and the scheduled start.	63
5	Selecting participating students; the generated access token is visible.	63
6	Deployment progress reaching 100%, indicating all student VMs are ready.	63
7	Live topology of a selected student with on-demand administrative actions (power off, suspend, and connect).	64
8	Terminal activity inside the VM: commands are logged to <code>commands.log</code> and recorded with Asciiinema.	64
9	Real-time progress chart with attempts, hints, and completed tasks per student.	65
10	Optional Grafana query filtering logs from <code>commands.log</code>	65
11	After finishing, instructors review time, score, penalties, and solved challenges.	66
12	Pop-up with the student’s command history for after-action review.	66

List of Tables

1	Overview of Exclusions and Reasons in Both Screening Phases	32
2	Coverage of Core and Emergent Modules Across Included Studies	42
3	Overview of Cyber Range Telemetry Sources, Grouped by System-Level and User-Level	52

List of Listings

1	Terraform Snippet Tagging a VM With Scenario/User IDs	47
2	Ansible Playbook Snippet Reporting a Web Service Deployment	47
3	Excerpt From a CALDERA JSON Log Showing Execution of a Technique	48
4	Example xAPI Statement: Participant Ran Out of Hints	49
5	Terraform Snippet Provisioning Observator and Student VMs on vSphere	56
6	Ansible Playbook Deploying Loki, Grafana and Optional Prometheus on the Observator VM	57
7	Docker Compose Configuration for Loki and Grafana on the Observator VM	58
8	Ansible Playbook to Configure Student VMs With Logging Tools	58
9	master_wrapper.sh: Start Ascinema Then Invoke the User Shell . .	60
10	commands_logger.sh: Record Each Command to commands.log .	61
11	Fluent Bit Config: Tail and Send Data to Loki	61

Definitions

Adaptive Difficulty System: Mechanism that dynamically adjusts the challenge level of a cyber range scenario in real time based on trainee performance or system conditions.

AI-Driven Orchestration: Use of artificial intelligence to manage and adapt cyber range resources, attacks, or scenarios automatically.

Cyber Range: Isolated, realistic environment designed to simulate complex cyber systems for cybersecurity testing, research, and training.

Data Exploitation: The analysis and modeling of collected data (logs, metrics, user actions) to derive insights, improve security, and enhance training outcomes.

Distributed Tracing: Technique for correlating events and logs across multiple microservices or system components to reconstruct the path of requests or attacks.

ELK Stack (Elasticsearch, Logstash, Kibana): A popular open-source suite for log aggregation, indexing, and visualization.

Experience API (xAPI): A flexible specification (originally from the e-learning domain) for capturing and storing events in a structured “actor–verb–object” format.

Federated Cyber Range: A cyber range setup that connects multiple independent ranges or sector-specific platforms to form a larger, collaborative environment.

Fluentd: Open-source data collector that unifies the ingestion of logs from diverse sources, applying transformations and forwarding them to a storage backend.

Gamification: In the cyber range context, the inclusion of scoring systems, leaderboards, badges, and other playful elements to increase motivation and engagement.

Grafana Loki: A logs aggregation system designed to index logs by labels, inspired by Prometheus’s metrics-focused design.

Infrastructure (in a Cyber Range): The foundational computing layer (physical or virtual) and associated platforms (VMs, containers, networks) hosting the simulated environment.

Monitoring: Continuous observability framework that aggregates logs, metrics, and events across a system for real-time awareness and alerting.

Orchestration: The automated deployment, scaling, and lifecycle management of services, virtual machines, or containers within a cyber range.

PRISMA (Methodology): A structured approach for conducting systematic literature reviews, ensuring transparent reporting of inclusion/exclusion criteria and search strategies.

Prometheus: Open-source time-series database and toolkit designed for real-time metrics collection and alerting.

SIEM (Security Information and Event Management): Platforms (commercial or open-source) that aggregate logs and security events, applying correlation rules to detect threats or intrusions.

Ansible: Open-source automation tool for configuration management and software deployment.

Asciiinema: Utility that records terminal sessions for later playback in a text-based format.

Containerization (Docker): Technique of packaging applications with all dependencies into isolated containers, often managed by Docker.

FastAPI: Python web framework for building high-performance APIs with automatic OpenAPI documentation.

Fluent Bit: Lightweight log processor and forwarder used to ship data from VMs or containers to a central backend.

Hypervisor: Software layer that allows multiple virtual machines to run on a single physical host by abstracting hardware resources.

Infrastructure-as-Code (IaC): Practice of provisioning and managing infrastructure through machine-readable configuration files rather than manual processes.

Kubernetes: Open-source system for automating deployment, scaling, and management of containerized applications.

MAPE-K Loop: Autonomic computing model consisting of Monitor, Analyze, Plan, Execute steps with a shared Knowledge base.

Microservices Architecture: Design approach that structures an application as a collection of small, loosely coupled services communicating over APIs.

MongoDB: Document-oriented NoSQL database storing data in flexible JSON-like structures.

Terraform: HashiCorp tool that defines and provisions infrastructure using declarative configuration files.

Telemetry: Continuous collection of metrics, logs, or traces describing the behavior of systems and users.

Virtual Machine (VM): Software-based emulation of a physical computer running its own operating system.

VMware vCenter: Suite for centralized management of VMware-based virtualization infrastructures.

List of Acronyms and Abbreviations

AI	Artificial Intelligence.
AIT	Austrian Institute of Technology.
API	Application Programming Interface.
APT	Advanced Persistent Threat.
AWS	Amazon Web Services.
AWX	Ansible AWX.
CEF	Common Event Format.
CLI	Command Line Interface.
CRATE	Cyber Range And Training Environment.
DARPA	Defense Advanced Research Projects Agency.
DevOps	Development and Operations.
DSL	Domain-Specific Language.
ECHO	European network of Cybersecurity centres and competence Hub for innovation and Operations.
ECISO	European Cyber Security Organisation.
ELK	Elasticsearch, Logstash, Kibana.
ESDL	ECHO Service Description Language.
FCR	Federated Cyber Range.
HCI	Human-Computer Interaction.
HEAT	OpenStack Orchestration Service.
HMI	Human-Machine Interface.
HR	Heart Rate.
HRV	Heart Rate Variability.
ICS	Industrial Control Systems.
IoT	Internet of Things.
JSON	JavaScript Object Notation.
LRS	Learning Record Store.
MAPE	Monitor, Analyze, Plan, Execute.
MITRE	MITRE Corporation.
ML	Machine Learning.
NLP	Natural Language Processing.
NPC	Non-Player Character.
OT	Operational Technology.
OTel	OpenTelemetry.
PaaS	Platform as a Service.
PLC	Programmable Logic Controller.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
QEMU	Quick Emulator.
RBAC	Role-Based Access Control.
REST	Representational State Transfer.
RKE2	Rancher Kubernetes Engine 2.
RQ	Research Question.

RTU	Remote Terminal Unit.
SCADA	Supervisory Control and Data Acquisition.
SIEM	Security Information and Event Management.
SOC	Security Operations Center.
SSH	Secure Shell.
TTP	Tactics, Techniques, and Procedures.
VLAN	Virtual Local Area Network.
VM	Virtual Machine.
VPN	Virtual Private Network.
xAPI	Experience API.
YAML	YAML Ain't Markup Language.

1 Introduction

1.1 Context and Motivation

Recent surveys indicate that nearly two-thirds of universities already operate—or plan to deploy—cyber range platforms for hands-on cybersecurity education [1], while systematic reviews have catalogued between 44 and 102 documented cyber range implementations or studies worldwide [2, 3]. A *cyber range* is a secure, controlled environment that simulates real-world network infrastructures and cybersecurity scenarios for training, testing, and research. Digital transformation has spurred sophisticated platforms—such as the *SCORPION Cyber Range* [4]—which may follow monolithic, service-oriented, or microservices-based designs, each with distinct trade-offs in scalability, flexibility, resilience, manageability, and fidelity [5, 6].

While microservices are often favoured for their modularity and ability to scale in dynamic virtualised settings [6, 7], monolithic architectures remain viable when simplicity and low overhead are required. Managing clusters of hypervisors and numerous containerised services leads to challenges—such as VM sprawl, resource contention, and scattered logs—that require robust orchestration and real-time monitoring [8, 9].

Despite the abundance of logs, metrics, and user interactions, many cyber ranges rarely collect or exploit these data effectively. Integrating information from heterogeneous components—physical servers, virtual machines, containerised applications, identity management systems, and analytics engines deployed on-premises or in the cloud—demands careful normalisation and correlation to detect anomalies and enhance training outcomes.

To address these requirements, our stack combines enterprise virtualisation with modern DevOps tooling. VMware vCenter manages hosts and VMs, FastAPI microservices backed by MongoDB deliver the range’s RESTful APIs and business logic, and Terraform with Ansible automates deployment and configuration, forming an agile foundation for real-time monitoring.

1.2 Challenges with Dynamic and Real-Time Data

Cyber ranges produce dynamic, multi-layered data that pose unique challenges in terms of collection, normalisation, and correlation. Distributed systems generate logs from various sources (e.g., Docker logs, Kubernetes events, database metrics, front-end usage events), each with its own format and timing. Detecting anomalies—such as performance degradations, malicious activities, or potential cheating incidents—requires robust, scalable solutions that can handle large data volumes with low latency [7]. Additionally, consolidating and analysing heterogeneous data streams in near real time is critical for advanced analytics and effective decision-making. Yet, most existing cyber ranges fail to fully exploit these data streams—highlighting a key opportunity for improvement through unified monitoring and analytics.

Another critical concern is ensuring stealth monitoring—using logging methods that remain undetectable to participants—in adversarial or competitive scenarios [10, 11]. Some

participants may try to disable or bypass logging to conceal their activities. Designing out-of-band monitoring components—e.g., port mirroring on switches or hypervisor-level introspection agents—helps maintain data integrity. The time synchronisation across all monitored systems is also essential for accurate event correlation. Small clock discrepancies—minor differences in system time between machines—can complicate the reconstruction of a timeline, especially when analysing multi-stage attacks that unfold within seconds [2].

1.3 Objectives of the Work

The primary objective of this work is to propose a conceptual framework for real-time monitoring and analysis of cyber range systems, highlighting modern architectural trends and deployment approaches. Specifically, this work aims to:

1. **Map the State of the Art:** Survey existing cyber range architectures, including recent trends and deployment modalities, to contextualise the building blocks required for real-time monitoring.
2. **Propose an Extensible Reference Architecture:** Outline a flexible, technology-agnostic design that can be applied across different cyber range implementations.
3. **Develop an Integrated Typology for Data Generation, Monitoring, and Exploitation:** Propose a comprehensive classification framework for diverse data sources (e.g., VM commands, container logs, performance metrics) and outline strategies for advanced analytics, including anomaly detection, sequence analysis, and predictive modelling [8, 4].
4. **Validate in a cyber range context:** The SCORPION Cyber Range serves as a partial validation of our approach, chiefly exercising the monitoring and data exploitation features rather than providing a full implementation of the entire proposed architecture and typology.

1.4 Contributions

While the objectives outline the steps and scope of the work, the key contributions of this study are:

- **Proposal of a Conceptual Cyber Range Architecture:** We present a generic architecture that integrates enterprise virtualisation with modern DevOps and monitoring tools for cyber range environments.
- **Integrated Data Typology:** We introduce a comprehensive classification for the data generation, the monitoring, and the exploitation of information, unifying heterogeneous data sources through standard formats (e.g., the Experience API (xAPI)) and Fluentd-based pipelines.

- **Practical Validation through SCORPION:** By applying our framework to the SCORPION Cyber Range, the work demonstrates how real-time data analytics can improve training effectiveness, operational resilience, and overall system scalability.

1.5 Document Organisation

This paper is organised as follows:

- **Section 1 (Introduction):** Provides the context, motivation, and main contributions of this study.
- **Section 2 (Literature Review and State of the Art):** Reviews the trends, architectural paradigms, deployment modalities, and standardisation considerations for modern cyber ranges.
- **Section 3 (Methodology):** Describes the systematic screening and coding approach used to fulfill the first objective.
- **Section 4 (Proposal of a Conceptual Cyber Range Architecture):** Presents a technology-agnostic model with distinct layers, detailing component interactions for comprehensive monitoring.
- **Section 5 (Typology for Data Generation, Monitoring, and Exploitation):** Defines how data are generated across layers, how they are captured (including xAPI), and how they can be mined for insights.
- **Section 6 (SCORPION Cyber Range Use Case):** Demonstrates the practical application of the proposed architecture and typology, highlighting specific scenarios, metrics, and analytical insights.
- **Section 7 (Conclusions):** Summarises the main findings and outlines avenues for future research and development.

2 State of the Art

2.1 Definition and Purpose of Cyber Ranges

Cyber ranges are *virtual environments* that replicate real-world networks, systems and threat scenarios in a safe, isolated setting for training, testing and research. The European Cyber Security Organization (ECSO) defines a cyber range as “a platform for the development, delivery and use of interactive simulation environments” that mirror an organization’s IT and OT infrastructure, including simulated users and adversaries [12]. In practice, a range allows students, professionals and military units to exercise defensive or offensive skills without risking live assets. The concept, originally rooted in government and defence programmes such as the National Cyber Range, now permeates academia and industry as the demand for hands-on cybersecurity experience grows [2, 13].

A range delivers *active learning*: participants experiment with tools, tactics and procedures while instructors observe repeatable, realistic scenarios. This “train-as-you-fight” philosophy makes cyber ranges a cornerstone of workforce up-skilling [8]. Moreover, ranges provide a proving ground for emerging defences (e.g. new IDS or response playbooks) before production deployment [6]. Consequently, they bridge the gap between theoretical knowledge and practical cyber-physical operations.

2.2 Classification of Cyber-Range Platforms

Modern literature distinguishes four canonical types of cyber ranges, each balancing fidelity, cost, and flexibility [14]. Simulation ranges are constructed entirely with virtual machines or containers and remain easy to (re)configure and highly scalable, making them popular for education and capture-the-flag events, albeit with limited physical realism [2]. Overlay ranges layer virtual scenarios on top of live infrastructure, integrating real devices for higher realism (e.g. GENI overlays) while raising cost and safety concerns [14]. Emulation ranges employ dedicated hardware to reproduce target environments one-to-one, yielding very high fidelity at significant setup cost [14]. Finally, hybrid ranges combine simulation, overlay, and emulation to leverage the strengths of each, a common choice for industrial control or IoT scenarios [15, 16].

Simulation excels in scalability and reset capability, whereas emulation and overlay are preferred when operational realism is paramount. Hybrid approaches increasingly deliver integrated solutions, and taxonomy can further consider domain focus (e.g. IT vs. Industrial Control Systems (ICS)) and deployment model (on-premises, cloud or federated) [17, 18].

2.3 Key Features and Emerging Trends

Recent surveys highlight a decisive evolution from monolithic laboratory test-beds to modular, data-driven *cyber range ecosystems* that emphasise scalability, adaptability, and learning analytics [8, 5]. Core trends are summarised below.

2.3.1 Virtualisation and Microservice Platforms

Hypervisor-based virtual machines and containers remain the foundation for range deployment. Recent platforms rely on infrastructure-as-code to automate complete lifecycles. Guerrero *et al.*'s *Tectonic* shows how academic courses can launch VMware or AWS instances on demand [19], while Luise *et al.* demonstrate per-student ephemeral ranges in the cloud [20].

In parallel, ranges increasingly adopt *microservice* architectures. The reference model by Kampourakis *et al.* advocates modular services orchestrated by Kubernetes, and industrial deployments report similar patterns [6, 21]. SCORPION exemplifies this design: its microservices deliver the analytics pipeline, gamification engine, and a topology generator that can automatically build scenarios composed of routers, host templates, and optional Docker services. The resulting scenarios may be refined and stored in a catalogue for later assignment to challenges rather than generated on-the-fly [4]. Such compositions enhance scalability but raise observability challenges addressed later in Section 2.5.

2.3.2 Federation and Interoperability

Federating multiple ranges allows organizations to pool specialised assets and conduct joint exercises. The EU ECHO project pioneers a *Federated Cyber Range* (E-FCR) with brokered access, capability catalogues and common APIs [22, 21]. Interoperability efforts include standard scenario-description formats [23] and federated identity management. Ear *et al.* categorise federation maturity models and argue that “Cyber Range 2.0” will require automated mapping of exercise demands to federated supply [24].

2.3.3 Gamification and Adaptive Orchestration

Game mechanics—points, badges, and leaderboards—consistently boost learner engagement [25]. The Cyber Arena platform described by Stodulka *et al.* and SCORPION both include native scoring engines and medal systems [26, 4]. Beyond static scripts, ranges increasingly support *dynamic or customisable scenarios*. SCORPION's authoring tool can rapidly generate candidate topologies, letting instructors spin up fresh scenarios on demand; once refined, these are catalogued for reuse, enabling dynamic orchestration of varied challenges [4]. Katsantonis *et al.* introduce a coach/agent that adapts difficulty based on real-time performance [27]. Such orchestration combines DevOps automation with narrative “injects” to mimic evolving incidents.

2.3.4 Monitoring and Data Exploitation

Comprehensive data pipelines aggregate system logs, network flows, and user events. Open-source stacks (Prometheus, ELK, Grafana Loki) are popular choices [28, 29, 30]. Stodulka notes many ranges still do not fully exploit their data, relying on manual after-action review [26]; research therefore pushes towards automated learning analytics and feedback loops [31]. SCORPION goes further by ingesting biometric heart-rate telemetry

and computing real-time stress metrics [4]. We examine user monitoring and infrastructure observability next.

2.4 User Behaviour Monitoring and Learning Analytics

Capturing fine-grained participant actions enables formative assessment and adaptive coaching. Bash DEBUG traps, keyloggers like `logkeys`, and terminal recorders such as `Asciinema` collect shell commands and full sessions for replay [32, 33, 4]. Events are often serialised as xAPI statements for cross-tool analytics [34, 31]. Sequence mining then extracts metrics such as completion paths, hint usage, and error frequency.

Gamification engines subscribe to these event streams, granting points or badges when triggers fire [26]. SCORPION’s analytics additionally fuse physiological signals to infer stress, allowing the scenario to scale difficulty up or down and providing novel research data on learner affect [4]. Such multimodal analytics promise personalised cyber training, though privacy and sensor accuracy remain open questions. Additionally, SCORPION automatically adjusts challenge resources—time limits, number of attempts, and available hints—based on each participant’s historical performance within the competition.

2.5 Infrastructure Observability in Microservice-Based Ranges

Microservices scatter logs, metrics, and traces across many containers and hosts. A “single pane of glass” therefore relies on aggregated observability stacks—e.g. Fluent Bit→Elasticsearch + Kibana, or Prometheus→Grafana [28]. Distributed tracing via OpenTelemetry links events end-to-end, enabling instructors to replay how an exploit propagated through services [35, 36]. Research highlights challenges such as high cardinality metrics and trace explosion, but also the opportunity to feed anomaly-detection models that alert when infrastructure or scenario behaviour deviates from expected patterns [7, 37].

Self-healing orchestration (e.g. Kubernetes auto-restarts or horizontal pod autoscaling) further increases platform resilience. When combined with rich telemetry, instructors maintain a “unified dashboard providing a comprehensive view” of exercise health, while automated scoring systems can verify whether defensive objectives (e.g. blocking exfiltration traffic) were met.

Conclusion

The contemporary cyber range landscape is defined by cloud-backed virtualisation, microservice modularity, federation, gamified adaptive scenarios, and data-driven monitoring. Exemplars such as SCORPION showcase how these facets converge: a microservice range delivering pre-built scenarios—crafted manually or generated with its topology tool—together with gamification and real-time learning analytics, including biometric stress detection, all orchestrated through IaC [4]. As AI-driven adversaries, immersive

3-D environments, and autonomous orchestration mature, cyber ranges are poised to become ever more realistic, adaptive and insightful, ultimately producing better-prepared cybersecurity practitioners.

3 Methodology

3.1 Scope and Methodological Approach

A *cyber range* is an isolated, realistic environment designed to simulate complex cyber systems for cybersecurity testing, research, and training. The methodology presented here establishes the process used to fulfill the first objective of this work: proposing a general architecture for cyber ranges through a systematic literature review.

3.2 Identification of Research Works and Initial Screening

Inspired by the PRISMA methodology [38], we began with a Scopus query targeting the terms “*cyber range*” and “*architecture*” within the 2020–2025 publication window. This search yielded 42 records, of which 2 were duplicates, leaving 40 unique items. An *initial screening* was performed as follows:

1. **Exclusion by Title and Abstract:** We removed 9 papers that either mentioned “cyber range” only tangentially or lacked any reference to relevant architectural details (e.g., focusing solely on threat modelling, mobile security specifics, or user behaviour emulation without describing an overarching CR architecture).
2. **Exclusion by Full Text:** We then examined the remaining 31 papers in more detail. 5 of these were excluded due to insufficient coverage of architectural components (e.g., focusing strictly on AI-based scenario management or purely educational outcomes without technical depth).

After the initial screening, 26 papers remained. A closer look showed that these 26 aligned better with our goal of identifying common modules in cyber ranges, as they generally included *some* architectural perspective—such as infrastructure layers or orchestration workflows—even if they varied in detail.

3.3 Refined Exclusion Criteria

We then applied a *refined* set of exclusion criteria to the remaining 26 papers, ensuring that each final inclusion closely addressed our interest in microservices-based architectures, real-time monitoring, and training-oriented capabilities. Specifically, works were removed if they:

- (a) Focused only on scenario building or domain-specific simulation (e.g., digital twins, purely IoT/vehicle contexts) without describing broader microservices or CR architecture components.
- (b) Presented only a superficial mention of “cyber range” without providing an architecture diagram or meaningful details on key modules (infrastructure, orchestration, monitoring, etc.).

- (c) Did not address training-focused features (e.g., user management, real-time monitoring, gamification) that connect directly to our analysis objectives.

Based on these refined criteria, *7 more papers* from the 26 were excluded.

- **Criterion (a)** accounted for the majority of these exclusions (e.g., papers dedicated to scenario generation, DSL-based approaches, or digital twin simulations *without* CR architectural depth).
- **Criterion (b)** removed a smaller subset that mentioned a “cyber range” but failed to present any tangible diagrams or identify modules (e.g., monitoring, virtualisation, orchestration).
- **Criterion (c)** eliminated a few studies that discussed architectures in a limited sense but offered no insight on real-time user training/monitoring or gamification.

In total, reducing from 26 to 19 papers left us with those that clearly detail infrastructure or virtualisation, orchestration, and at least 1 layer of training-oriented capabilities (e.g., user management or logging analytics). We also manually added 1 extra relevant paper—*our own work on the SCORPION Cyber Range*—bringing the final total to 20 papers.

3.4 Detailed Rationale for Exclusions

Table 1 provides an overview of the main reasons for exclusion across both the initial and refined screenings, summarizing the reasons for exclusion in each phase as outlined below:

In particular:

- *Secure Infrastructure for Cyber-Physical Ranges* [39], *P-I2Prange* [40], or *Cyber Arena* [41] were excluded due to **(b)**—no substantial diagrams or clear architecture components.
- *AI-assisted Computer Network Operations testbed...* [37] focused heavily on **(a)**, i.e., scenario building or advanced simulations but minimal mention of key CR modules.
- Works like *Using Augmented Reality for Learning Naval Operations* [42] or *Analysis of Cyber-Attacks for Modern Digital Railway System* [43] were also excluded for **(a)/(c)**—they targeted specialized scenarios without describing training-oriented CR architecture.

3.5 Including an Additional Relevant Paper

Outside of the Scopus-based query, we *manually included our own* SCORPION Cyber Range paper [4]. Despite not appearing in the original search results, it significantly contributes to our topic by:

Table 1: Overview of Exclusions and Reasons in Both Screening Phases

Phase	Reasons for Exclusions (Examples)
Initial Screening (40 → 26)	• <i>Title/Abstract:</i> Papers focusing on <u>threat modelling</u>, specific <u>mobile security tasks</u>, or <u>user emulation</u> without describing CR architecture (9 excluded). • <i>Full Text:</i> Papers lacking detailed CR components; e.g., purely <u>AI scenario management</u> or <u>educational frameworks</u> with minimal technical detail (5 excluded).
Refined Criteria (26 → 19)	• (a) Scenario-only or digital-twin focus, no broader CR modules: e.g., specialized DSL or domain-specific simulations. • (b) No architecture diagram or insufficient detail on main components: e.g., mentioning “cyber range” but not specifying virtualisation, orchestration, or monitoring layers. • (c) Lack of training-oriented aspects: e.g., no user management, real-time monitoring, or gamification needed for our architectural analysis. At this stage, 7 more papers were removed in total.

- Describing a microservices-based cyber range that integrates gamification and biometric monitoring.
- Showcasing real-time data exploitation strategies.
- Aligning well with the goal of identifying common components and orchestration approaches.

Consistent with PRISMA guidelines, we include it as an “additional record identified through other sources.” This raises our final total to 20 papers.

3.6 Inclusion Criteria and Final Corpus

Ultimately, each of the 20 included papers addresses at least 1 of the following domains:

1. Infrastructure and virtualisation strategies (*e.g.*, VMs, containers, or hybrid setups).
2. Orchestration workflows (Terraform, Kubernetes, OpenStack, etc.).
3. User management (multi-team support, gamification, red/blue teaming).
4. Monitoring capabilities (metrics, logging, dashboards).
5. Data exploitation or analytics (scoring, feedback, learning analytics).
6. Emergent themes (AI-driven orchestration, IoT integration, federation, ICS components).

This process ensures that only studies providing a meaningful view of cyber range architectural components, training-oriented modules, or microservices-based designs are retained.

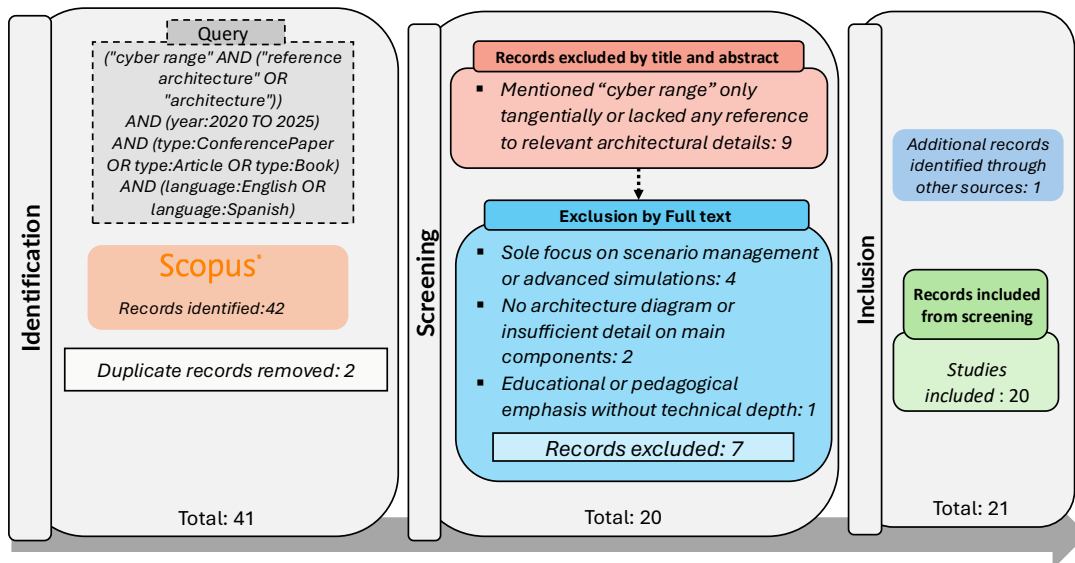


Figure 1: PRISMA Flow Diagram Adapted for the Systematic Review of Cyber Range Architectures (2020–2025)

3.7 Review and Coding Process

A *combined deductive–inductive* coding methodology was employed:

1. **Deductive Coding:** Based on established cyber range literature, we defined 5 main components (*infrastructure, orchestration, user management, monitoring, data exploitation*).

2. **Inductive Coding:** We recorded any newly observed features (e.g., AI-based orchestration, IoT, ICS integration, advanced gamification).
3. **Keyword Extraction:** Terms such as “Kubernetes,” “OpenStack,” “digital twin,” and “learning analytics” were used to capture technical specifics.
4. **Comparative Synthesis:** Finally, we prepared a comparative matrix highlighting how each of the 20 works addresses the core components and emergent themes.

This methodology reflects a two-stage exclusion process—initial (focusing on overall relevance and minimal architectural content) and refined (assessing deeper alignment with training, microservices, and real-time monitoring)—while also accommodating an extra paper of particular relevance not found via Scopus.

The resulting survey serves as the foundation for a generic reference architecture and an integrated telemetry typology encompassing data generation, monitoring, and the exploitation of information.

4 Proposal of a Conceptual Cyber Range Architecture

4.1 Overview

This subsection presents the findings of our review of 20 cyber range architectures published between 2020 and 2025. We begin with concise summaries of each included work (§4.2), highlighting how authors address five core (deductive) components—*infrastructure*, *orchestration*, *user management*, *monitoring*, *data exploitation*—as well as additional (inductive) modules uncovered during coding (e.g., *federation services*, *advanced scenario orchestration*, *gamification engine*, *biometric data*, *multi-cloud support*, *policy and compliance integration*). We then provide comparative tables (§4.3) enumerating which components each paper implements, followed by a discussion of emergent features and a proposed extended reference architecture (§4.4).

4.2 Summaries of Included Papers

Below are short synopses of the 20 papers included in our final corpus. Each paragraph outlines the paper’s principal objectives, core architecture, and notable features related to both the five foundational components (infrastructure, orchestration, user management, monitoring, data exploitation) and any emergent modules. Each entry lists the paper by title and citation without numbering or inclusion in the table of contents.

The ECHO Project (E-FCR) [22]

Zanasi *et al.* (2021) introduce a federated architecture connecting multiple cyber ranges. The ECHO Federated Cyber Range (E-FCR) interlinks multiple provider-specific ranges. Their microservices-based design includes an *Access Portal*, *Broker Gateway*, *QoS Manager*, and other components, illustrating orchestration and federation capabilities that coordinate multiple providers. A *Capacity/Capability Map* tracks sector-specific services (e.g., Industrial Control Systems (ICS), healthcare). The E-FCR user management handles roles and billing, while an analytics engine captures data for usage and performance feedback, moving toward integrated data exploitation. This work demonstrates multi-sector infrastructure integration and a modular approach.

Tectonic Cyber Range [19]

Guerrero *et al.* (2024) present an academic range built with a full Infrastructure-as-Code pipeline. Tectonic offers infrastructure either on AWS or on-premises. Orchestration relies on Packer, Terraform, and Ansible, together with a Python-based scenario controller. Monitoring is handled via the *Elastic Security* stack, while data exploitation is partially supported through SIEM analytics. A planned integration with Moodle highlights emphasis on user management and bridging theoretical content with hands-on

labs. Though not explicitly federated, it underscores multi-cloud and repeated scenario creation.

On-Demand Cloud Cyber Ranges [20]

Luise *et al.* (2021) enable on-demand AWS-based ranges through automated orchestration. They propose a dynamic, AWS-based infrastructure that can spin up ephemeral “Macro Ranges,” subdivided into “Micro Ranges” for different labs. An *Orchestration* layer automates creation and termination, with real-time monitoring and security checks enforced by a *Cluster Security Controller*. User management is handled via AWS Cognito. Although it does not feature advanced data exploitation, the approach is strongly oriented toward multi-tenant usage and real-time isolation. Their architecture highlights an early step toward multi-cloud support.

Cyber Arena [26]

Stodulka *et al.* (2022) implement an OpenStack range with integrated scoring and ICS hardware. “Cyber Arena” is orchestrated with Heat templates and AWX/Ansible, addressing infrastructure provisioning and orchestration. The *Training Application* manages user roles, scoring, and gamified logic, while monitoring leverages Prometheus, Grafana, and Elastic. There is limited data exploitation beyond basic logging; however, bridging real ICS hardware with the virtual environment is novel. The approach does not feature explicit federation, but it uses microservices patterns.

Hybrid IoT Cyber Range [15]

Balto *et al.* (2023) blend physical IoT devices with virtual nodes for repeated experiments. They combine physical IoT devices (cameras, sensors) with emulated nodes in QEMU, showcasing infrastructure complexity. Provisioning uses Ansible, Terraform, and OpenStack Ironi, indicating advanced orchestration. Monitoring is conducted via ELK, and device resets/“flashing” are automated for repeated experiments. Although user management is mentioned, the principal focus is on realistic IoT/OT labs. Future expansions may include more comprehensive data exploitation, with partial log correlation already in place.

Frankenstack [44]

Pihelgas *et al.* (2021) provide a real-time log-correlation and feedback system. They describe a distributed log-correlation pipeline for monitoring and near real-time feedback. By ingesting Sysmon, Suricata, network flows, and applying Sigma correlation, data exploitation is prominent. While the paper does not detail infrastructure or user management extensively, it contributes an advanced monitoring framework. This approach can integrate with larger orchestrations and provide comprehensive detection feedback for multi-actor exercises.

SCADA Cyber Range for Active Learning [16]

Cruz *et al.* (2021) build a hybrid ICS/SCADA training environment with basic automation. They focus on Industrial Control Systems/Supervisory Control and Data Acquisition (ICS/SCADA) training with a hybrid infrastructure: real PLCs and motors together with virtual networks. Their architecture includes user management for guided exercises and basic monitoring for ICS protocols; orchestration is currently handled through deployment/reset scripts, with further automation proposed as future work. Data exploitation is mostly limited to logging; however, bridging physical ICS with ephemeral VMs fosters realistic hands-on training.

ECHO Federated Cyber Range (Extended) [21]

Oikonomou *et al.* (2021) detail cross-provider federation via an ECHO mid-tier platform. They provide deeper insight into the E-FCR, describing federation services bridging multiple ranges, an ESDL (ECHO Service Description Language) for scenario definitions, and a mid-tier microservices platform for brokered requests, QoS, and billing. Orchestration, infrastructure, user management, and partial data exploitation appear in multiple subsystems (e.g., analytics, usage logs). This paper further highlights how distinct providers can unify under a single large-scale scenario.

High-Fidelity 5G-based IIoT Testbed [45]

Cruz *et al.* (2023) demonstrate a Kubernetes-based 5G testbed for industrial IoT. They present a Kubernetes-based testbed for industrial IoT over 5G networks, implementing infrastructure via container orchestration (Rancher RKE2) and orchestration workflows for microservices-based ICS solutions (OpenPLC). Basic monitoring occurs through the cluster's standard dashboards, though advanced data exploitation is minimal. User management is not the main focus, yet the testbed showcases how 5G, together with ICS modules, can be integrated for next-generation industrial security exercises.

Industrial Control Cyber Range System [46]

Low *et al.* (2022) offer a VM-based ICS range with manual orchestration and attack tools. They design a small-scale ICS range using VMware Workstation with vulnerable OS images. The authors emphasize infrastructure for ICS tasks, manual orchestration (start/stop snapshots), and integrated attack-defense modules (Metasploit). Monitoring involves logging attacker/defender actions, while data exploitation is generally limited. The environment aims at replicable ICS vulnerability exploitation training.

DefAtt Virtual Cyber Labs [25]

Mahmoud *et al.* (2021) introduce gamified virtual labs using Docker and VirtualBox. They implement a *Game Engine* for scenario setup/teardown, a *WireGuard Engine* for secure VPN access, and an *Open vSwitch Engine* for dynamic VLAN creation—all under orchestration. Docker/VirtualBox-based infrastructure supports containerized or VM-based services. A “Virtual SOC” with Elastic Stack and TheHive underscores monitoring and data exploitation, while a gamified multi-user approach addresses user management.

CyberIoT [47]

Sharifi *et al.* (2021) propose a web-based IoT range with sandbox management on OpenStack. Their three-tier platform supports IoT training with OpenStack provisioning. The *Sandbox Management* module orchestrates resources, focusing on infrastructure and orchestration for IoT labs. User management supports multi-tenant concurrency, while the monitoring module tracks resource usage. Although data exploitation remains limited, the paper emphasizes how IoT resources can be rapidly configured as “sandboxes.”

CyberBOSS [48]

Welch *et al.* (2020) coordinate multiple simulators via a broker for LVC&G exercises. CyberBOSS acts as a *broker* for multi-simulator integration. The “Actor Correlation Manager” and “Delegation Manager” handle cross-platform orchestration and device identity alignment, underlining a form of federation for bridging different tools or simulators (e.g., OneSAF, COBWebS). While the paper focuses less on monitoring or data exploitation, it illustrates a high-level infrastructure approach for “fused” LVC&G (Live, Virtual, Constructive, and Gaming) exercises.

CR-ICS (Industrial Control Systems) [49]

Khan *et al.* (2021) present a hybrid IT/OT environment with Python-based orchestration. They introduce CR-ICS, a hybrid IT–OT environment featuring an *AnyLogic* simulator for industrial processes connected to containerized or VM-based ICS services. Orchestration employs Python scripts and open vSwitch for dynamic segmentation. The monitoring plane includes traffic mirroring, and a control plane manages scenario deployment and user sessions, addressing user management. Data exploitation is mainly partial logs of ICS anomalies and normal user actions.

Addressing the IoT Security Gap [18]

Nock *et al.* (2020) suggest a modular architecture for IoT cyber ranges. They survey IoT security testbeds and propose a modular architecture for physical–virtual integration.

While the paper does not detail orchestration pipelines extensively, it covers the need for an API-based approach. Basic user management and monitoring exist, with an emphasis on *REST endpoints* to coordinate IoT labs. They stress infrastructure flexibility and mention the possibility of integrating with larger range frameworks, hinting at federation readiness.

AIT Cyber Range [50]

Leitner *et al.* (2020) describe an OpenStack-based platform with a scenario engine. The range is built on OpenStack, Terraform, and Ansible, forming an orchestration pipeline. A “Scenario Engine” goes beyond basic deployments, orchestrating timed attacks, messages, and resource reconfiguration. Monitoring is typically delegated to available OpenStack or external dashboards, and data exploitation is possible via logs. User management supports multi-team training, though advanced gamification is not described.

Cyber Range Design Framework [27]

Katsantonis *et al.* (2023) provide a lifecycle-based design with adaptive feedback. They describe a structured CR design framework with modules for workspace deployment, tracking participant actions, a coach for adaptive feedback, and a scenario repository. This addresses infrastructure, advanced orchestration, detailed monitoring, user management, and continuous data exploitation (e.g., performance analytics). The authors emphasize a lifecycle approach: analysis → design → configuration → deployment → dry run → execution. Their architecture is particularly effective at providing real-time feedback and dynamic scaffolding, an emergent form of *advanced scenario orchestration*.

SCORPION Cyber Range [4]

Nespoli *et al.* (2024) offer containerized exercises with gamification and learning analytics. They propose a container-based infrastructure with Terraform deployments for fully customizable cyber exercises. A *Gamification Engine* awards points and medals, integrated with a *Learning Analytics System* that collects telemetric and biometric data (heart rate), supporting data exploitation for stress detection. The scenario authoring tool can randomize topologies, illustrating advanced scenario orchestration. User management integrates with adaptive difficulty adjustments, distinguishing SCORPION as a next-generation approach.

Automated Cyber Range Design (Reference Architectures) [24]

Ear *et al.* (2023) map organisational requirements to six deployment archetypes. They identify six reference architectures (physical, centralized virtual, private cloud, public cloud, distributed virtualization, and hybrid) for cyber ranges. A Python-based matching algorithm aligns 22 organisational requirements with these infrastructure archetypes,

focusing on cost, fidelity, and scalability. Although orchestration details are abstracted, the study addresses policy and compliance factors (e.g., security, legal constraints) as a key dimension. Monitoring and user management are generically covered, while data exploitation is less emphasized.

Model-Driven Deployment (CTL) [51]

Braghin *et al.* (2020) use a domain-specific language to generate OpenStack templates. They present a model-driven approach with a Domain Specific Language called CTL (Cybersecurity Training Language), translating high-level designs into OpenStack HEAT templates. This focuses on infrastructure automation and orchestration via tooling that reduces configuration errors. While user management, monitoring, and data exploitation remain more limited, the approach leverages formal scenario definitions for consistent deployments, which can be extended to advanced scenario orchestrations.

4.3 Comparative Tables

To illustrate how each study addresses the established five components and the newly identified modules, we provide a consolidated comparison in Table 2. Rows represent the 20 included papers (labeled by first author or short reference), and columns capture:

1. The five *deductive* components: *Infrastructure (Infra.)*, *Orchestration (Orch.)*, *User Management (UM)*, *Monitoring (Monit.)*, *Data Exploitation (DataExp.)*
2. The primary *inductive* (emergent) modules: *Federation Services*, *Advanced Scenario Orchestration*, *Secrets Management*, *Gamification Engine*, *Biometric Data*, *Multi-Cloud Support*, *Policy & Compliance*.

Table icons indicate coverage: ✓ denotes presence, ~ indicates partial support, and ✗ marks absence or minimal coverage. Brief text lists the relevant technology or feature. Common abbreviations (e.g. “VM,” “Docker,” “OpenStack,” “Ansible,” “Terraform”) appear in the table, and some module names are shortened for space.

Table 2: Coverage of Core and Emergent Modules Across Included Studies

Paper	Deductive Components					Emergent Modules						
	Infra.	Orch.	UM	Monit.	DataExp.	Fed.	Adv.Orch.	Sec.Mgt.	Gamif.	Bio.	Multi-Cl.	Policy
Zanasi et al. [22]	Multi-CR	Microservices	Roles/Billing	Basic metrics	Analytics engine	✓	~	✗	✗	✗	Potential	✗
Guerrero et al. [19]	AWS/On-prem	Terraform, Ansible	Moodle (planned)	Elastic SIEM	Basic SIEM	✗	✗	✗	✗	✗	✓	✗
Luise et al. [20]	AWS-based	Automated (AWS)	Cognito	Real-time checks	✗	✗	✗	✗	✗	✗	~	✗
Stodulka et al. [26]	OpenStack	Heat + Ansible	Roles, scoring	Prometheus, Grafana	✗	✗	✗	✗	Basic	✗	✗	✗
Balto et al. [15]	Hybrid IoT	Ansible, Terraform	Basic	ELK logs	✗	✗	✗	✗	✗	✗	✗	✗
Pihelgas et al. [44]	✗	✗	✗	Kafka, Suricata	Sigma correlation	✗	✗	✗	✗	✗	✗	✗
Cruz et al. [16]	ICS hybrid	Scripts	Basic roles	ICS logs	Minimal	✗	✗	✗	✗	✗	✗	✗
Oikonomou et al. [21]	Multi-CR	E-FCR mid-tier	RBAC	Basic metrics	Usage logs	✓	~	✗	✗	✗	✗	✗
Cruz et al. [45]	Kubernetes (5G)	RKE2	Basic	Cluster dashboards	✗	✗	✗	✗	✗	✗	✗	✗
Low et al. [46]	VMware	Snapshots	✗	Attack logs	Minimal	✗	✗	✗	✗	✗	✗	✗
Mahmoud et al. [25]	Docker/VM	Game Engine	Multi-user	Elastic, TheHive	✓	✗	✗	✗	~	✗	✗	✗
Sharifi et al. [47]	OpenStack	Sandbox Mgr	Multi-tenant	Basic usage	✗	✗	✗	✗	✗	✗	✗	✗
Welch et al. [48]	Multi-sim	Broker	✗	✗	✗	~	✗	✗	✗	✗	✗	✗
Khan et al. [49]	Hybrid IT/OT	Python scripts	Team mgmt	Traffic mirror	Minimal	✗	✗	✗	✗	✗	✗	✗
Nock et al. [18]	Physical+Virtual IoT	REST APIs	Basic	Packet capture	✗	Potential	✗	✗	✗	✗	✗	✗
Leitner et al. [50]	OpenStack	Terraform+Ansible	Multi-team	Basic logs	✗	✗	~	✗	✗	✗	✗	✗
Katsantonis et al. [27]	Cloud/Container	Automated (CWMS)	Multi-user	Live tracking	Adaptive feedback	✗	✓	✗	✗	✗	✗	✗
Nespoli et al. [4]	Docker	Terraform	Multi-user	Logs + Telemetry	Learning Analytics	✗	✓	✗	✓	✓	✗	✗
Ear et al. [24]	6 archetypes	Generic	Generic	Generic	Minimal	✗	✗	✗	✗	✗	~	✓
Braghin et al. [51]	OpenStack	DSL to HEAT	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Note on ~ (partial support) – The yellow curved dash marks functionalities that are *present but not fully implemented*. Examples: (i) Welch [48] connect multiple simulators via a broker, achieving limited federation; (ii) Balto [15] offer basic log correlation, not a full analytics pipeline; (iii) Luise [20] deploy only on AWS while designing for multi-cloud. The same reasoning applies to every ~ cell.

4.4 Emergent Features and Final Conceptual Architecture

4.4.1 Emergent (Inductive) Modules

While the five deductive components (*infrastructure, orchestration, user management, monitoring, data exploitation*) dominate the landscape, our review highlights at least eight additional modules (see Table 2) that extend cyber range functionality:

- **Federation Services:** Present mainly in [22, 21, 48], enabling cross-platform range integration and resource sharing.
- **Advanced Scenario Orchestration:** Found in [27, 4], going beyond simple provisioning to provide dynamic attacks, timed injects, or adaptive event triggers.
- **Secrets Management:** A recognized need (though rarely explicit) for handling credentials and configuration data securely at scale.
- **Gamification Engine:** Illustrated by [25, 26, 4], awarding points, achievements, or levels to enhance trainee motivation.
- **Biometric Data / Telemetry:** Demonstrated most notably by [4], measuring stress or cognitive load in real time for adaptive feedback.
- **Multi-Cloud Support:** Increasingly relevant for deployments bridging AWS, Azure, or on-prem resources, seen in [19] and partially in [20].
- **Policy & Compliance Integration:** Addressed in general form by [24], signifying the rising importance of organisational governance, auditing, and regulatory alignment in range exercises.
- **External Services & Digital Twins (subset of Infrastructure Extension):** Some works ([15, 16]) integrate real hardware or ICS digital twins for advanced realism; others (e.g., [21]) connect distributed domain simulators.

Collectively, these emergent modules expand cyber ranges from purely technical “training labs” to integrated ecosystems that address operational complexity (federation, multi-cloud, secrets), advanced learning strategies (gamification, adaptive scenario control), and organisational requirements (policy compliance, advanced data analytics).

4.4.2 Proposed Extended Reference Model

Figure 2 conceptually summarizes how these newly identified modules integrate with the original five-component structure. In particular, we further refine two key modules often combined under *gamification* and *advanced scenario* design:

- **Game Engine Subcomponent:** Manages scoring, points, or badges, and can generate tokens to access specific VMs or unlock challenges. This component interfaces with:
 1. *User Management*: to authenticate students, manage teams and roles, and keep consistent user records.
 2. *Data Exploitation*: to retrieve performance metrics and logs for dynamic scoring or awarding achievements.
- **APT/NPC Simulator Subcomponent** (within *Advanced Scenario Orchestration*): Generates adversarial or “NPC” (non-player character) traffic to simulate either malicious or benign behaviours. This subcomponent:
 1. Integrates with the Orchestration Layer to inject dynamic attacks or user-like events at specific intervals.
 2. Feeds logs into *Monitoring* and *Data Exploitation* to assess how effectively trainees detect or mitigate threats.

This additional granularity clarifies how realistic threat simulations and game-driven user progression coexist under the broader architecture. Ultimately, these subcomponents allow a Cyber Range to integrate timed attack injections, stealthy adversaries, or reward systems based on user performance, aligning with both educational and operational goals.

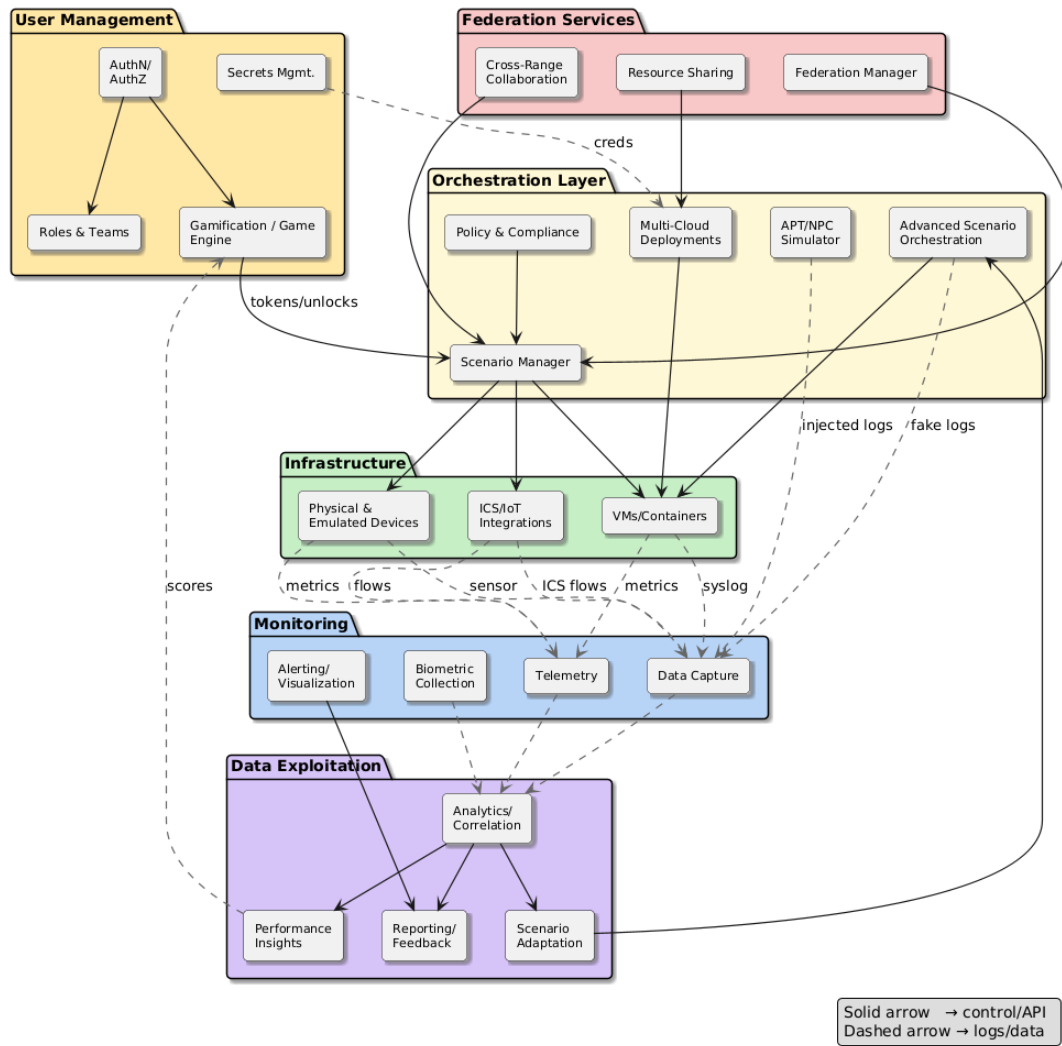


Figure 2: Illustrative Extended Reference Model Integrating the Five Pillars and Emergent Modules

By layering these new modules alongside the established five, modern cyber ranges can more effectively address educational goals, multi-site collaboration, and real-world threats. The trend toward microservices and domain-specific expansions (ICS/SCADA, IoT, 5G, healthcare) highlights how next-generation cyber ranges are evolving into flexible, federated platforms that integrate both the core pillars and advanced features such as gamification, biometric monitoring, multi-cloud deployments, and integrated policy management.

5 Typology for Data Generation, Monitoring, and Exploitation

Building on the architecture introduced in the previous chapter (Industrial Control Systems/Operational Technology (ICS/OT) integration, federation of environments, and multi-cloud orchestration), this chapter presents a *typology* of telemetry data in the cyber range, covering generation, monitoring, and exploitation of that data. We distinguish two broad categories of telemetry: system-level data (emitted by the infrastructure, automation, and simulated adversaries) and user-level data (produced by participant interactions and team dynamics).

This section explains how these diverse logs are normalised into common formats (e.g., Common Event Format (CEF), OpenTelemetry [35], Experience API (xAPI) [34]) and correlated through shared keys such as *userID*, *teamID*, or *scenarioID*. Raw Tactics, Techniques, and Procedures (TTP) logs from adversaries (run with tools like MITRE CALDERA) and infrastructure traces are transformed into these normalised formats. The impact of specific architectural features (Industrial Control Systems/Supervisory Control and Data Acquisition (ICS/SCADA) integration, range federation, and multi-cloud deployments) on the telemetry strategy is highlighted as well.

Throughout the chapter, code snippets (Terraform, Ansible, CALDERA, xAPI) illustrate data generation and analytics integration. Finally, advanced analytics and scenario adaptation are discussed, detailing how telemetry powers Monitor, Analyze, Plan, Execute (MAPE) feedback loops and real-time stream processing pipelines (Apache Spark/Flink) to achieve scenario self-adaptation. Table 3 at the end of the chapter summarises the complete data typology. This holistic view of data generation, monitoring, and information exploitation highlights the fundamental role of telemetry in enabling dynamic cyber training, setting the stage for the “SCORPION” use case introduced in the next chapter.

5.1 System-Level Data Representation

System-level telemetry includes all data emitted by the cyber range infrastructure and automated scenario components (provisioning, configuration, simulated adversaries, etc.). This data reflects the state and behaviour of the environment independently of user actions. The main sources of system-level telemetry—shaped by the federated, multi-cloud architecture—are described below:

5.1.1 Provisioning Logs (Terraform)

Infrastructure-as-Code tools like Terraform generate detailed traces when creating and configuring resources (e.g., virtual machines (Virtual Machines (VMs)), networks). Instrumenting these tools to include scenario-related metadata in their output (e.g., `ScenarioID` and `UserID` tags) greatly facilitates subsequent correlation of cloud or virtualisation events. Listing 1 shows a simplified Terraform snippet tagging a VM with `scenario_id` and `user_id`:

Listing 1: Terraform Snippet Tagging a VM With Scenario/User IDs

```
1 resource "aws_instance" "web" {
2   ami          = "ami-0abc12345def67890"
3   instance_type = "t2.micro"
4
5   tags = {
6     Name          = "Scenario-${var.scenario_id}-WebServer"
7     ScenarioID    = var.scenario_id
8     UserID        = var.user_id
9   }
10 }
```

By parsing Terraform’s `plan/apply` output or its `state` file, these logs can be exported to CEF or OpenTelemetry (OTel) [35], allowing real-time monitoring of infrastructure creation. Subsequent events (VM launches, cloud alerts) inherit scenario/user metadata, easing correlation and filtering (e.g., “all alerts for Scenario 7”).

5.1.2 Configuration & Scenario Execution Logs (Ansible)

After infrastructure is provisioned, system configuration or scenario events are launched with tools like Ansible. Each *playbook* logs detailed information (timestamps, target hosts, success/failure), allowing scenario progression to be tracked. These logs are integrated into the telemetry pipeline by parsing Ansible’s standard output or adding hooks (e.g., Representational State Transfer (REST) calls to a telemetry Application Programming Interface (API) or xAPI statements). Listing 2 shows how the playbook reports the deployment to the telemetry API:

Listing 2: Ansible Playbook Snippet Reporting a Web Service Deployment

```
1 - hosts: webserver
2   vars:
3     scenario_id: "{{ scenario_id }}"
4   tasks:
5     - name: Deploy web application
6       become: yes
7       apt:
8         name: apache2
9         state: present
10
11     - name: Register deployment event
12       uri:
13         url: "http://telemetry.example/api/scenario/{{
14           scenario_id }}/events"
15         method: POST
16         body: >
17           {
18             "event": "webserver_deployed",
19             "host": "{{ inventory_hostname }}"
20           }
```

```
20         status_code: 200
```

Note the use of YAML’s multiline (`>`) syntax in the `body` field, which avoids any confusion with braces or quotes inside the `lstlisting` environment. Once normalised (e.g., to CEF or enriched JSON), these events can be correlated with other sources (CALDERA logs, system logs), yielding a coherent timeline of the scenario.

5.1.3 Adversarial Actions (MITRE CALDERA)

A key system-level telemetry component comes from adversary simulators like MITRE CALDERA. CALDERA automatically executes Tactics, Techniques, and Procedures (TTPs) via installed agents, producing JSON logs for each attack step (ATT&CK technique, target host, timestamp, outcome). These logs capture adversarial activity for later correlation. Listing 3 shows an example:

Listing 3: Excerpt From a CALDERA JSON Log Showing Execution of a Technique

```
1 {
2   "agent_metadata": {
3     "paw": "pertbn",
4     "host": "WORKSTATION1"
5   },
6   "attack_metadata": {
7     "technique_id": "T1005",
8     "technique_name": "Data from Local System"
9   },
10  "operation_metadata": {
11    "operation_name": "My Operation"
12  },
13  "status": 0,
14  "agent_reported_time": "2025-02-23T11:50:13Z"
15 }
```

Because CALDERA logs are already structured, the platform can convert them to CEF or transform them into xAPI statements (e.g., “Adversary agent ran T1005 on host X”), while retaining scenario IDs, hosts, timestamps, etc. This correlation is essential for linking adversary and defender activities.

5.1.4 ICS/OT Telemetry (Industrial Systems)

When the cyber range includes Industrial Control Systems (ICS) or Supervisory Control and Data Acquisition (SCADA) components, specialised Operational Technology (OT) data emerges. Industrial equipment — such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Human-Machine Interfaces (HMIs) — emits event logs, alarms, or process time-series (e.g., temperature or pressure). These signals are typically collected through protocol gateways or historians. For example, a water plant simulation may produce sensor logs and anomaly alerts. By matching device or process-

variable identifiers, cyber-physical incidents can be correlated with IT events (e.g., “malware found on an HMI” aligns with “anomalous water level reading”).

5.1.5 Cloud Platform Events (Multi-Cloud)

In multi-cloud scenarios, native logs from each provider (AWS CloudTrail, Azure Activity Logs, etc.) are also gathered, tagging each resource with `ScenarioID` and `UserID`. These logs are accessed via APIs and integrated into the platform. This allows detection of misconfigurations (e.g., insecure security groups) or injection of cloud-specific incidents that the participant must handle.

5.1.6 Federation Metadata (Multi-Range)

In distributed or federated ranges, the orchestration layer produces synchronisation logs (start/end of exercises, site confirmations). Each event is labeled with scenario and site IDs, enabling a global timeline across multiple locations. If one site lags behind, a MAPE loop can delay certain injects to maintain consistency.

5.2 User-Level Data Representation

User-level telemetry captures participant actions, behaviours, and states while interacting with the scenario. Main sources include:

5.2.1 xAPI Learning Experience Events

The Experience API (xAPI) [34] standard describes learning activities in `<actor, verb, object>` form. The following JSON document records when a participant runs out of hints:

Listing 4: Example xAPI Statement: Participant Ran Out of Hints

```

1 {
2   "actor": {
3     "name": "Trainee Bob"
4   },
5   "verb": {
6     "id": "http://example.com/expapi/verbs/ran_out_of_hints",
7     "display": { "en-US": "ran out of hints" }
8   },
9   "object": {
10    "id": "http://cyber.range/scenarios/SCORPION/Challenge7"
11  },
12  "result": {
13    "success": false,
14    "completion": false
15  },

```

```
16     "timestamp": "2025-05-27T12:34:56Z",
17     "context": {
18         "extensions": {
19             "scenarioId": "SCORPION",
20             "teamId": "Blue"
21         }
22     }
23 }
```

These statements are stored in a Learning Record Store (LRS), allowing correlation of learner progress (*completed*, *requested hint*, etc.) with system-level events (e.g., adversary attacks, configuration logs).

5.2.2 HCI Interaction Logs (Mouse, Keyboard, Command Line Interface (CLI))

Capturing low-level interaction (shell commands, keystrokes, mouse clicks, console sessions) provides granular insight into user strategies. Tools like *asciinema* [32] (terminal recordings) or *logkeys* [33] (keylogger) can be used. Likewise, libraries such as *WebGazer.js* [52, 53] enable webcam-based eye-tracking. All these logs are normalised (e.g., *OpenTelemetry* [35] or *xAPI* extensions) and correlated by *userID*, *scenarioID*, and *timestamps*.

5.2.3 Biometric and Physiological Signals

Wearable devices can provide heart rate (HR) or galvanic skin response, quantifying stress or cognitive load. These continuous data streams are integrated into the platform as time-series (e.g., *heart_rate=120bpm*) and can trigger adaptive actions (e.g., delaying a new challenge if HR exceeds a threshold).

5.3 Monitoring & Analytics

Once system-level and user-level telemetry is unified, we apply robust monitoring and advanced analytics:

5.3.1 Data Normalisation and Correlation

Given the variety of sources, events are converted into a few standard formats (CEF, *OpenTelemetry* [35], *xAPI*) and tagged with common identifiers (*scenarioID*, *userID*, *host*, *timestamp*). Logging pipelines (e.g., *Fluentd* or *Beats*) send events to a message bus (e.g., *Kafka*), where streaming engines (*Spark*, *Flink*) can process them in real time.

5.3.2 Advanced Analytics and Adaptation (MAPE Loops)

Inspired by autonomic computing, the MAPE-K model (Monitor, Analyse, Plan, Execute, with a Knowledge base) supports real-time adaptation of the scenario:

- *Monitor*: Continuous ingestion of telemetry (system logs, xAPI, ICS, biometrics).
- *Analyse*: Streaming processing in Spark/Flink, detecting patterns (e.g., attacks with no user response) or anomalies (e.g., high stress).
- *Plan*: Choosing adaptive actions (launch an extra attack, provide a hint, increase difficulty, alert the instructor).
- *Execute*: Orchestrating on-the-fly changes (e.g., Terraform/Ansible to add a new vulnerable host, or sending a hint message to the participant).

All executed adaptations are also logged, closing the feedback loop.

5.4 Summary of the Data Typology

Table 3 summarises the main telemetry data sources, how each is generated, monitored, and correlated, and what adaptation opportunities they enable:

Data Source	Usage	Adaptation	Linkage	Standards
System-Level Sources				
Provisioning Logs (Terraform)	Created during VM/network resource setup.	Validate deployment, retry failures, dynamically adjust topology.	Resource IDs (VMs, networks), <i>scenarioID</i> , <i>userID</i> .	Terraform JSON, normalised to CEF/OpenTelemetry.
Configuration Logs (Ansible)	Produced by each <i>playbook</i> task.	Fix errors, ensure correct order of service start, sequence <i>injects</i> .	Hostname, Ansible task, <i>scenarioID</i> .	Ansible output parsed, converted to CEF or xAPI.
Adversarial Actions (CALDERA)	TTPs mapped to ATT&CK with timestamps, success/fail.	Escalate attacks if user is too efficient, or pause them otherwise.	host, technique_id, <i>scenarioID</i> .	Native JSON, exported to CEF/xAPI.
ICS/OT Telemetry	Logs from PLCs/R-TUs, process simulations, alarms, time-series.	Trigger safety or anomaly protocols, measure response to physical changes.	Device IDs, <i>scenarioID</i> , timestamps.	Extended CEF for OT, OpenTelemetry for metrics.
Multi-Cloud Events (AWS, Azure, etc.)	Cloud-native API/security/audit logs.	Simulate cloud outages, scale resources on demand, detect misconfigs.	Account ID, region, resource ID, <i>scenarioID</i> .	Proprietary JSON mapped to CEF/OTel.
Federation Metadata (Multi-Range)	Synchronisation between sites, phase start/end, handshake messages.	Delay global injects until all sites are aligned, maintain consistent timeline.	<i>scenarioID</i> , <i>siteID</i> .	Proprietary JSON, normalised to CEF/OTel.
User-Level Sources				
xAPI (Learning Events)	Tracks task completions, hints used, achievements.	Adjust difficulty (launch more attacks if user is fast, provide help if stuck).	actor, verb, object, <i>scenarioID</i> .	xAPI (IEEE 9274-1:2022).
CLI/Keyboard Interaction (logkeys, asciinema)	Commands, keystrokes, shell history.	Detect repeated failures, confusion, or reaction to an attack.	<i>userID</i> , host, timestamp.	asciinema [32], logkeys [33]; normalised to CEF/OTel/xAPI.
Eye-Tracking, Mouse (WebGazer.js)	Mouse clicks, gaze position, scroll.	Alert if user ignores warnings.	<i>userID</i> , x, y, timestamp.	WebGazer [52, 53], xAPI for interaction.
Biometrics (Heart Rate, Stress)	Wearables providing HR or HRV signals, etc.	Reduce difficulty or trigger interventions if stress remains high.	<i>userID</i> , heart_rate, timestamps.	OTel metrics, possible xAPI extensions.
Team Communication (chat, voice)	Chat or audio logs for multi-player scenarios.	Detect confusion or blockages, prompt instructor to assist.	teamID, <i>userID</i> , timestamps.	Raw text logs or transcripts, Natural Language Processing (NLP) for sentiment analysis.

Table 3: Overview of Cyber Range Telemetry Sources, Grouped by System-Level and User-Level

5.5 Conclusion

This chapter has outlined a complete typology of telemetry in the cyber range, covering both system-level data (infrastructure logs, adversarial actions, ICS/OT, multi-cloud) and user-level data (xAPI, HCI interaction, biometrics). The discussion has shown how these heterogeneous data streams can be normalised (to CEF, OpenTelemetry, and xAPI) and correlated via shared context identifiers (`scenarioID`, `userID`, `teamID`, etc.). The chapter also discussed how to apply MAPE loops and streaming analytics (Spark/Flink) to dynamically adapt the scenario.

By unifying operational and behavioural data, telemetry forms the bridge between the technical execution of the environment and continuous learning assessment. The next chapter presents the “SCORPION” use case, demonstrating how this typology and adaptive strategies are realised in a concrete scenario, improving cyber-training effectiveness.

6 SCORPION Use Case

6.1 Introduction and Context

SCORPION is a microservice-based cyber-range platform that integrates gamified learning mechanics as a first-class design principle rather than an optional add-on. Training content is organised into *challenges* that belong to a *scenario*: a software-defined topology composed of virtual machines (VMs) and containerised services. Instructors can register their own VM templates or upload new ones, mapping them to different roles within the scenario.

Gamification permeates every layer of the system. Learners progress through challenges using configurable navigation modes—e.g. linear, survival, or mixed paths—with medals, badges, and live scoreboards providing continuous feedback. SCORPION also supports collaborative and adversarial exercises (red–blue or full red–blue–white formats), enabling both individual labs and team-based competitions.

Beyond traditional scoring, the platform captures rich telemetry—including per-command traces, network and system metrics, and optional biometric indicators such as heart-rate-derived stress levels. These data feed an observability stack—provisioned automatically via Ansible and Terraform and powered by Fluent Bit, Loki, and Docker—that enables real-time dashboards as well as post-exercise analytics (e.g. sequence mining, bottleneck detection, learner modelling).

Taken together, SCORPION offers an end-to-end environment where scenario orchestration, intrinsic gamification, and comprehensive monitoring coexist to support both skill acquisition and empirical research into cyber-range training effectiveness.

6.2 Key Technology Decisions

Open-source observability

We chose Loki + Grafana for logs over heavier stacks such as *ELK* because:

- **Lightweight:** Loki indexes only metadata (labels), reducing memory/storage overhead.
- **Integration:** Grafana natively supports Loki, providing user-friendly dashboards for both logs and metrics.
- **Cost:** All components are free, open-source, and well-documented. There are no licensing constraints.

Choosing Fluent Bit

Logstash or Beats could do similar forwarding, but Fluent Bit is written in C, has a much smaller resource footprint on each Student VM, and includes a native Loki output plugin.

Avoiding a commercial SIEM

Traditional SIEMs (e.g., Splunk, QRadar) are powerful but often expensive, with features (correlation rules, threat intelligence) beyond the scope of a student lab. The open-source stack is sufficient and more customisable for academic use.

Why Asciinema

Standard shell history or system logs do not preserve interactive context (such as text editors, error prompts, or interactive menus). Asciinema provides a time-based replay, letting instructors watch exactly what the student saw. The recorded sessions are stored in lightweight JSON, which is straightforward to archive or retrieve later.

6.3 Design

6.3.1 Overall Architecture

In SCORPION, an *instructor* chooses which cyber exercise to run, selects the participating students or teams, and optionally enables navigation or gamification rules. The platform’s microservices—such as a lifecycle manager and an orchestrator—then deploy the scenario automatically, creating all required infrastructure:

- One *Observer VM* (an observer VM that centralizes monitoring).
- Multiple *Student VMs* (one per participant or team).

As part of the deployment, the system reports progress and assigns each VM to the appropriate student or team.

Once deployed, each student VM is provisioned with:

1. A mechanism to capture *all shell commands* via a Bash DEBUG trap and custom logging scripts.
2. An Asciinema-based recording system that starts automatically upon user login.
3. A lightweight log forwarder (Fluent Bit) to stream logs and session files to the Observer VM.

The Observator VM runs a containerised Grafana + Loki stack (optionally Prometheus for metrics). This instance implements the monitoring module of our conceptual framework. Although we leverage Grafana for advanced filtering and personalised dashboards, the primary SCORPION interface remains its own web-based platform, ensuring that critical analytics and log views are integrated into a single pane for instructors.

6.3.2 Data Collection Workflow

When a student logs in (either by SSH or graphical interface), a `master_wrapper.sh` script executes:

1. Starts an Ascinema recording (saving the session to `/var/log/ascinema/...`).
2. Launches a shell that has a `DEBUG` trap to log every command to `/var/log/commands.log`.

As commands and terminal output are recorded, Fluent Bit tails those log files (in JSON format) and streams them to the Observator's Loki service in near real time.

6.4 Implementation

This section outlines the implementation of the SCORPION Use Case, assuming the following prerequisites:

- A virtualization environment (e.g., vSphere) where Terraform can provision VMs.
- Ansible to configure both the Observator VM and the Student VMs.
- A Git repository where all configuration files and scripts are stored.

The key configuration files reside in a `code/` directory, and their main contents are summarised below.

6.4.1 Terraform Snippet

Terraform defines resources for both the Observator and multiple student machines. An excerpt appears in `code/terraform-vsphere-excerpt.tf` (Listing 5):

Listing 5: Terraform Snippet Provisioning Observator and Student VMs on vSphere

```

1 # Provider and other resources omitted for brevity
2
3 resource "vsphere_virtual_machine" "student_vms" {
4   for_each = var.student_machines
5   name     = each.value.name
6   clone {
7     template_uuid = data.vsphere_virtual_machine.template_ubuntu.id
8     customize {
9       linux_options {

```

```

10     host_name = each.value.name
11     domain   = each.value.domain
12 }
13 network_interface {
14     ipv4_address = each.value.ip_address
15     ipv4_netmask = each.value.netmask
16 }
17 ipv4_gateway = each.value.gateway
18 }
19 }
20 }
21
22 resource "vsphere_virtual_machine" "observer" {
23     name      = var.observer_machine.name
24     num_cpus  = var.observer_machine.cpu
25     memory    = var.observer_machine.memory
26     # Similar customization as student_vms
27 }

```

After Terraform completes, the resulting IP addresses are passed to Ansible, which installs and configures the monitoring components.

6.4.2 Ansible Playbook (Observer)

Listing 6 presents the playbook used to configure the Observer VM:

Listing 6: Ansible Playbook Deploying Loki, Grafana and Optional Prometheus on the Observer VM

```

1  ---
2  - name: Configure Observer VM (Docker stack)
3    hosts: "{{ target_host }}"
4    become: yes
5
6    tasks:
7      - name: Install Docker & docker-compose
8        apt:
9          name:
10            - docker-ce
11            - docker-compose
12          state: present
13
14      - name: Copy docker-compose.yml
15        copy:
16          src: "{{ tools_path }}/docker-compose.yml"
17          dest: /opt/monitoring/docker-compose.yml
18
19      - name: Launch stack
20        shell: |
21          cd /opt/monitoring
22          docker-compose up -d
23          # Additional configuration omitted

```

This playbook copies a `docker-compose.yml` (see Listing 7) and relevant config files to `/opt/monitoring/` on the Observer. It then launches Grafana, Loki, and optionally Prometheus containers.

6.4.3 Docker Compose (Observer)

Listing 7 shows the Docker Compose file defining the container services:

Listing 7: Docker Compose Configuration for Loki and Grafana on the Observer VM

```

1 version: "3.3"
2
3 services:
4   loki:
5     image: grafana/loki:2.0.0
6     ports:
7       - "3100:3100"
8     volumes:
9       - loki-data:/loki
10    command: [ "-config.file=/mnt/config/loki-config.yml" ]
11
12   grafana:
13     image: grafana/grafana:latest
14     ports:
15       - "3000:3000"
16     volumes:
17       - grafana-panels:/etc/grafana/provisioning/dashboards/
18         panels
19 volumes:
20   loki-data:
21   grafana-panels:

```

After running `docker-compose up -d`, Loki listens on port 3100 for log ingestion, and Grafana (on 3000) provides the web dashboard.

6.4.4 Ansible Playbook (Student VMs)

Listing 8 illustrates the playbook used to configure each Student VM:

Listing 8: Ansible Playbook to Configure Student VMs With Logging Tools

```

1 ---
2 - name: Configure Student VMs
3   hosts: all
4   become: yes
5
6   tasks:
7     - name: Install packages

```

```

8     apt:
9         name:
10             - asciinema
11             - td-agent-bit
12         state: present
13
14     - name: Copy monitoring scripts
15       copy:
16         src: "{{ tools_path }}/observability-config/{{ item
17 }}}"
18         dest: "/usr/local/bin/{{ item }}"
19         mode: '0755'
20       loop:
21         - commands_logger.sh
22         - master_wrapper.sh
23
24     - name: ForceCommand in sshd
25       blockinfile:
26         path: /etc/ssh/sshd_config
27         block: |
28             Match User *
29                 ForceCommand /usr/local/bin/master_wrapper.sh
30       notify: restart_ssh
31
32     - name: Create log locations
33       file:
34         path: "{{ item.path }}"
35         state: "{{ item.state }}"
36         mode: "{{ item.mode }}"
37       loop:
38         - { path: /var/log/commands.log, state: touch, mode:
39 '0666' }
40         - { path: /var/log/asciinema, state: directory, mode:
41 '0777' }
42
43     - name: Configure Fluent Bit
44       template:
45         src: "{{ tools_path }}/observability-config/td-agent-
46 bit.conf.j2"
47         dest: /etc/td-agent-bit/td-agent-bit.conf
48       notify: restart_fluentbit
49
50 handlers:
51     - name: restart_ssh
52       service: { name: ssh, state: restarted }
53     - name: restart_fluentbit
54       service: { name: td-agent-bit, state: restarted }

```

52 # Additional setup omitted for brevity

This playbook:

- Installs Asciiinema for terminal session recording.
- Installs Fluent Bit to forward logs to the Observator's Loki.
- Deploys the `commands_logger.sh` and `master_wrapper.sh` scripts.
- Configures `/etc/ssh/sshd_config` so that every SSH login is wrapped by `master_wrapper.sh`.

6.4.5 Shell Wrapper and Command Logger

The playbook references several scripts. For example:

code/master_wrapper.sh

(Listing 9), the entrypoint script for any login shell:

Listing 9: `master_wrapper.sh`: Start Asciiinema Then Invoke the User Shell

```

1  #!/bin/bash
2  # master_wrapper.sh -- record session with asciiinema, then exec bash
3  .
4  set -euo pipefail
5
6  # Avoid recursion
7  [[ -z "${OBSWRAPPER_RAN:-}" ]] || exec /bin/bash --login "$@"
8  export OBSWRAPPER_RAN=1
9
10 # Ensure TTY
11 tty -s || exec /bin/bash --login "$@"
12
13 ts=$(date +%Y%m%d_%H%M%S)
14 user=$(whoami)
15 prefix=$(( [[ -n ${SSH_TTY:-} ]] && echo "ssh" || echo "local")
16 dir=/var/log/asciiinema
17 file="${dir}/${prefix}_${user}_${ts}.json"
18
19 # Reset flag so the command logger loads anew
20 unset OBSLOGGER_RAN
21
22 # Launch asciiinema or fallback
23 if command -v asciiinema >/dev/null 2>&1; then
24     exec asciiinema rec -y -q --command "/bin/bash --login" "$file"
25 else
26     exec /bin/bash --login "$@"
27 fi

```

code/commands_logger.sh

(Listing 10), which sets a DEBUG trap to log every command in JSON:

Listing 10: `commands_logger.sh`: Record Each Command to `commands.log`

```

1 #!/bin/bash
2 # commands_logger.sh -- log each command via trap DEBUG to /var/log/
   commands.log
3
4 [[ $- == *i* && -t 0 ]] || return 0
5 [[ -z ${OBSLOGGER_RAN:-} ]] || return 0
6 export OBSLOGGER_RAN=1
7
8 log_cmd() {
9     local cmd=$BASH_COMMAND
10    [[ -z $cmd ]] && return
11
12    # Simple skip patterns
13    [[ $cmd == history* ]] && return
14    [[ $cmd == printf* ]] && return
15    [[ $cmd == log_cmd* ]] && return
16
17    local ip=$(echo "${SSH_CLIENT:-}" | awk '{print $1}')
18    local ts=$(date -u +'%Y-%m-%dT%H:%M:%S.%3NZ')
19
20    local json_entry=$(jq -c -n \
21        --arg hostname "${hostname}" \
22        --arg timestamp "$ts" \
23        --arg ip "${ip:-}" \
24        --arg cmd "$cmd" \
25        --arg wd "$(pwd)" \
26        --arg user "$(whoami)" \
27        '{hostname: $hostname, timestamp: $timestamp, ip: $ip, cmd: $cmd,
28         wd: $wd, user: $user}')
29
30    echo "$json_entry" >> /var/log/commands.log
31 }
32 trap log_cmd DEBUG

```

When these scripts are installed, every interactive shell session loads `commands_logger.sh`, while `master_wrapper.sh` ensures `Asciinema` runs from session start.

6.4.6 Fluent Bit Configuration

Finally, each Student VM needs a `td-agent-bit.conf` (Fluent Bit) that tails the logs, as in Listing 11:

Listing 11: Fluent Bit Config: Tail and Send Data to Loki

```

1 [INPUT]

```

```

2   Name  tail
3   Path  /var/log/commands.log
4   Tag   ssh_commands
5   Parser json
6
7 [INPUT]
8   Name  tail
9   Path  /var/log/asciinema/*.json
10  Tag   asciinema_logs
11  Parser json
12
13 [OUTPUT]
14  Name  loki
15  Host  10.10.15.100
16  Port  3100
17  labels job=ssh-commands,hostname=${HOSTNAME}

```

This snippet instructs Fluent Bit to match lines tagged as `ssh_commands` (the commands log), parse them as JSON, and route them to Loki at `$(LOKI_HOST):3100`. It also forwards Ascinema session files under a distinct label. These logs appear in near real time on both Grafana dashboards and the SCORPION web interface.

6.5 Use-Case Dry-Run

The following walkthrough narrates a typical instructor-led session. Each step is illustrated with screenshots that highlight the most relevant interface elements.

In this training catalogue (see Figure 3), available cyber-exercises appear with a `+` button. The instructor selects a previously prepared training and initiates the deployment workflow.

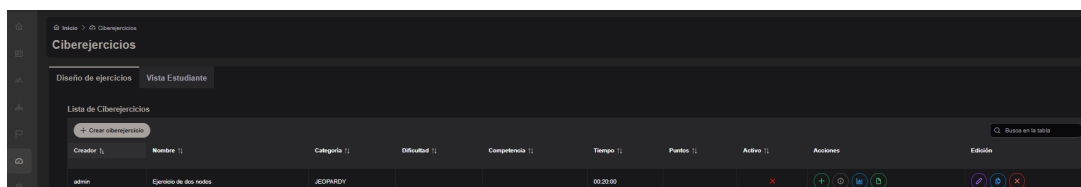


Figure 3: Training catalogue where the instructor clicks `+` to deploy an existing exercise.

As shown in Figure 4, the configuration modal lets instructors decide whether to enable guided navigation, activate difficulty adaptation, or schedule the start time. These options tailor the exercise to the students' skill level and availability.

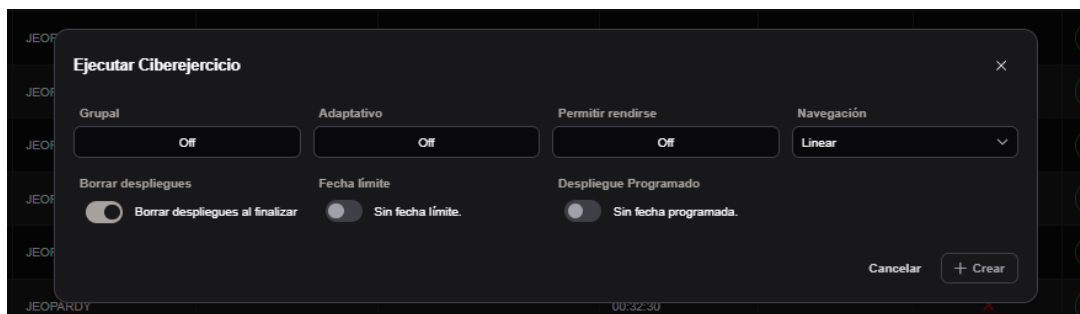


Figure 4: Modal for configuring navigation, adaptive mode and the scheduled start.

Next, as depicted in Figure 5, the instructor chooses the participants. The interface shows the unique access token that each student will use to connect to their VM.

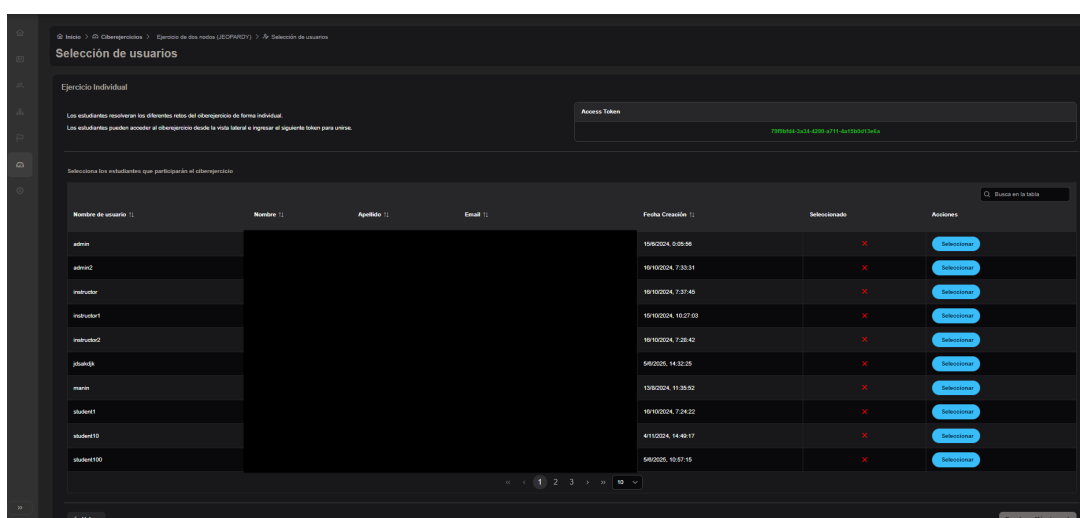


Figure 5: Selecting participating students; the generated access token is visible.

The platform then provisions the Observator VM and one machine per student, as illustrated in Figure 6. A progress bar confirms when everything is ready for the start of the exercise.

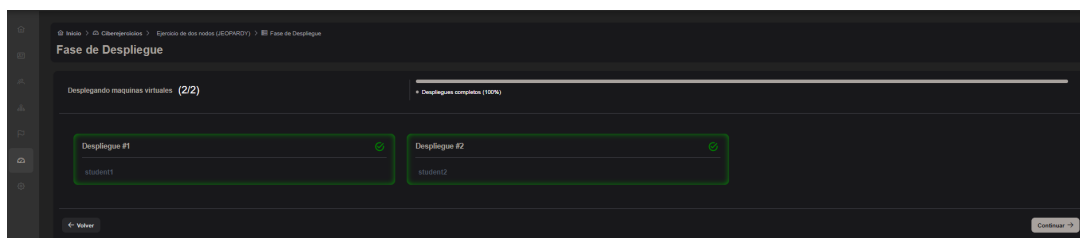


Figure 6: Deployment progress reaching 100%, indicating all student VMs are ready.

Once deployed, instructors can monitor each student's network diagram (see Figure 7). On-demand administrative actions allow them to temporarily power off or suspend a VM, introducing extra challenges or providing assistance on demand.

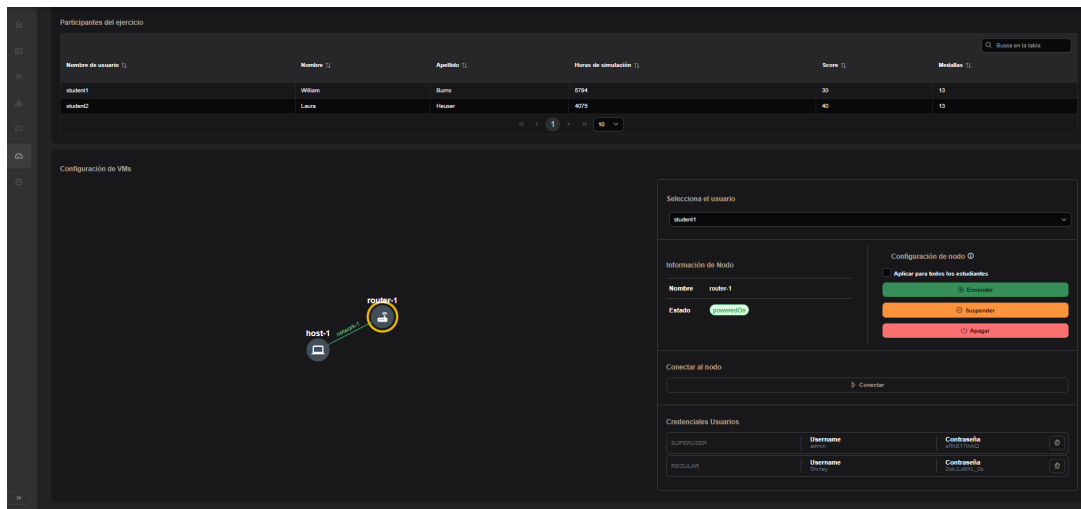


Figure 7: Live topology of a selected student with on-demand administrative actions (power off, suspend, and connect).

Figure 8 captures a student working in an Ubuntu desktop terminal inside the VM. Each command typed is logged and simultaneously recorded for later review.

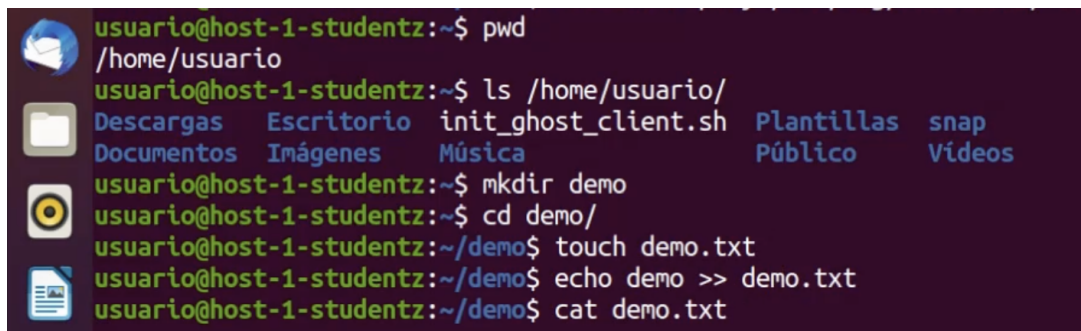


Figure 8: Terminal activity inside the VM: commands are logged to `commands.log` and recorded with Asciiinema.

Timeline markers, shown in Figure 9, indicate when hints were used or penalties accrued, helping instructors gauge how students approach each challenge.

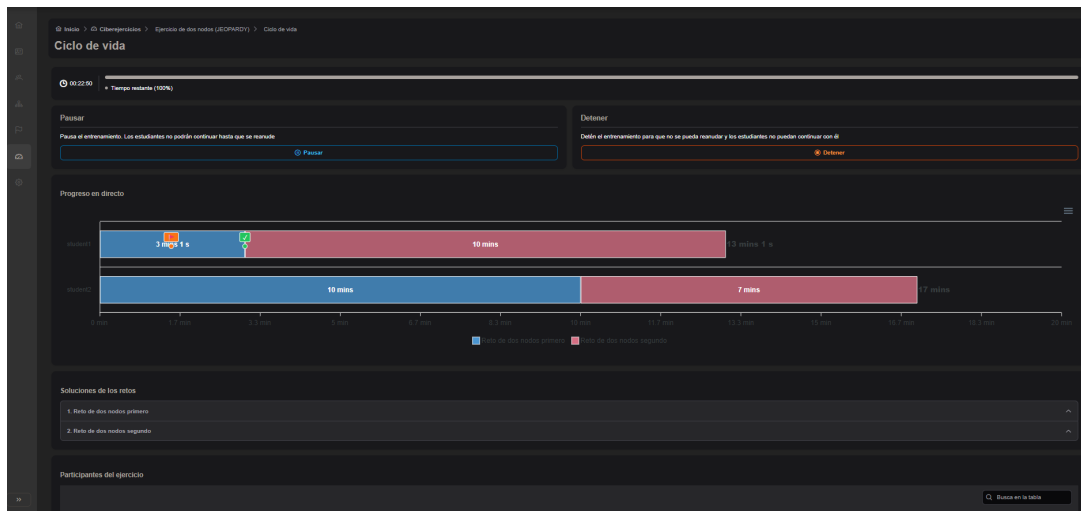


Figure 9: Real-time progress chart with attempts, hints, and completed tasks per student.

For advanced investigations, instructors can open Grafana and run Loki queries, filtering the same command logs to produce customised dashboards, as illustrated in Figure 10.

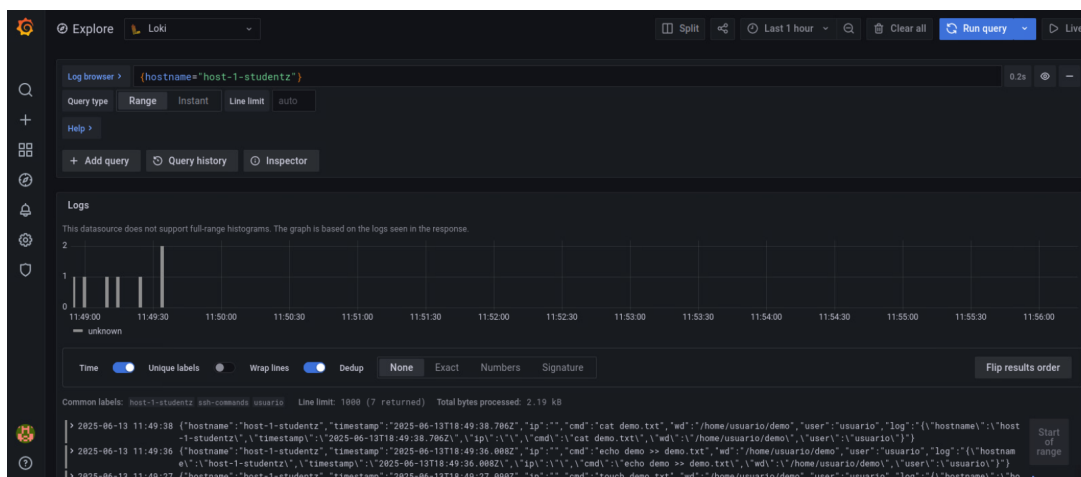


Figure 10: Optional Grafana query filtering logs from `commands.log`.

At the end of the exercise, SCORPION summarises key metrics for each student, accompanied by performance charts (see Figure 11).

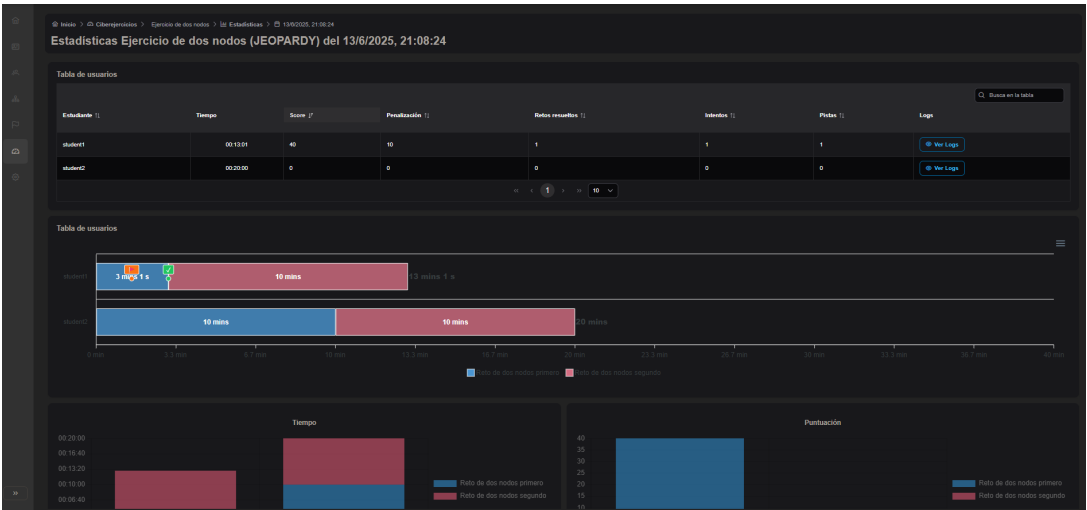


Figure 11: After finishing, instructors review time, score, penalties, and solved challenges.

The detailed command log can be inspected through a dedicated popup (Figure 12) that displays the timestamp, command, host, IP address, and working directory. Every command typed by the student is captured in real time and stored for later playback. Logs remain accessible during the session for immediate instructor review. Each session is stored as a lightweight JSON file produced by Asciiinema. These files can be replayed in any modern browser using a minimal player such as `asciiinema-player`—without installing Asciiinema—or in any terminal where the Asciiinema CLI is available, offering a realistic, time-accurate view of the student’s actions.

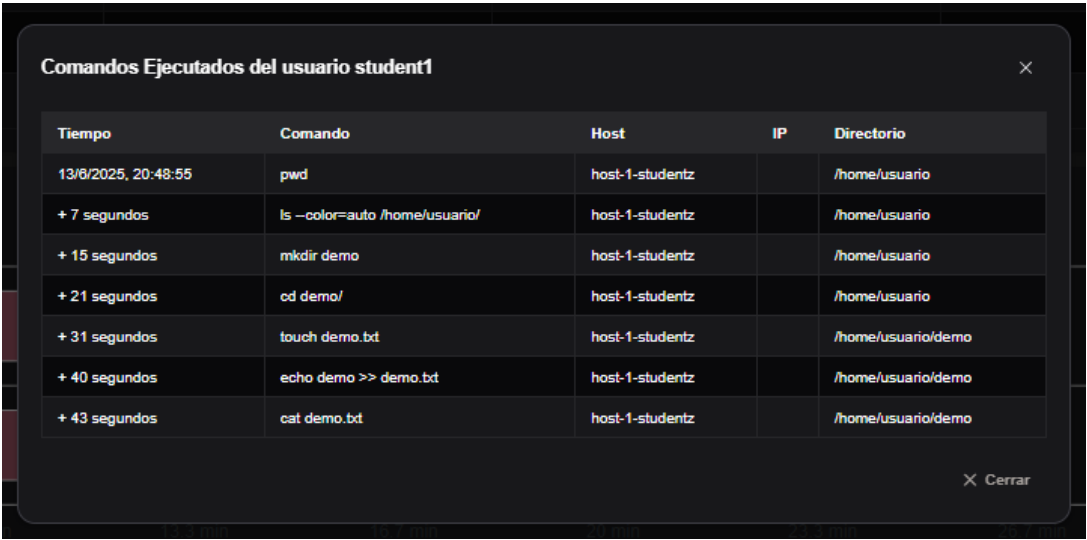


Figure 12: Pop-up with the student’s command history for after-action review.

Through these steps, SCORPION demonstrates how live monitoring and comprehensive logging support both immediate guidance and later assessment.

7 Conclusions

The present dissertation set out to confront the long-standing deficit of real-time observability in microservice-oriented cyber ranges, positing that pedagogical insight and operational resilience can only emerge when infrastructure telemetry and learner interactions are captured, correlated and acted upon in real time. Guided by four research objectives—mapping the state of the art, proposing an extensible reference architecture, devising a unifying telemetry typology, and validating both within the SCORPION range—the study has advanced the literature from descriptive surveys toward an integrated, evidence-driven paradigm of cyber range design.

A first theoretical contribution is the extended reference model that layers a dedicated monitoring tier—Fluent Bit to Loki and Grafana for logs, Prometheus for metrics and OpenTelemetry for traces—between orchestration and analytics. By elevating observability to a cross-cutting concern alongside infrastructure, user management, and data exploitation, the model resolves a blind spot documented in recent surveys, where monitoring is often relegated to post-exercise log scraping. The architecture further introduces emergent modules—federation services, advanced scenario orchestration, secrets management, gamification engines, biometric telemetry, multi-cloud elasticity, and policy compliance—that anticipate the complexity of next-generation ranges and offer a vocabulary for systematic comparison across academic and industrial deployments.

Equally important is the proposed telemetry typology, which normalises heterogeneous system-level streams (Terraform, Ansible, CALDERA, cloud audit events, ICS process data) and user-level streams (xAPI statements, shell recordings, keystrokes, gaze traces and physiological signals) into a coherent data fabric keyed by `scenarioID`, `userID` and temporal metadata. This abstraction enables real-time sequence mining, anomaly detection and adaptive orchestration through a MAPE-K feedback loop, thereby turning every packet, command and heartbeat into analysable evidence rather than archival noise.

Empirical validation within SCORPION demonstrates the feasibility and utility of the framework. Terraform instantiated an *Observer* node and per-student virtual machines on a vSphere cluster; Ansible installed containerised monitoring stacks; and lightweight Bash wrappers captured every learner keystroke without perceptible overhead. During small-scale pilot exercises, instructors observed live dashboards updating at sub-second granularity, replayed full terminal sessions, and resolved learner bottlenecks in minutes rather than hours. These anecdotal observations suggested that diagnostic latency fell sharply, misconceptions in command syntax surfaced early, and biometric stress trends informed pacing decisions, collectively illustrating that real-time insight is pedagogically transformative rather than a mere convenience.

From a practical standpoint, the study shows that enterprise-grade observability need not entail commercial SIEM licences or heavyweight ELK clusters. A carefully curated open-source stack—Fluent Bit, Loki, Grafana and optional Prometheus—delivers sufficient fidelity for instructional oversight while remaining deployable on modest academic hardware. Moreover, the Infrastructure-as-Code pipeline aligns with DevOps best practices, ensuring repeatable deployments, rapid resets and traceable configuration drift, all of which are essential for large-scale, multi-cohort teaching scenarios.

Pedagogically, the integration of gamification and learning analytics bridges a critical gap between technical instrumentation and learner motivation. SCORPION’s scoring engine, adaptive hints and stress-aware pacing showcase how telemetry can be repurposed from passive oversight to active coaching, fostering self-regulated learning and objective competence assessment. By mapping event streams to formal frameworks such as NICE or ENISA, the approach paves the way for automated skills accreditation and personalised learning paths that extend beyond the confines of a single exercise.

Nonetheless, several limitations temper the generalizability of these findings. The validation cohort was confined to one university context and predominantly Linux-based scenarios; Windows endpoints, cloud-native container orchestration (e.g., Kubernetes), and globally distributed learners remain to be stress-tested. The ethical and legal ramifications of continuous biometric monitoring warrant deeper scrutiny, particularly in workforce-development settings subject to stringent data-protection regimes. Federated deployments raise open questions in trust brokering, policy harmonisation and wide-area telemetry latency; and the current adversary model, while automated, lacks the adaptive intelligence of human red teams.

Future research should therefore pursue four avenues. First, extending Fluent Bit connectors to Windows Event Logs and Sysmon will close a platform gap and enable heterogeneous endpoint coverage. Second, integrating Terraform providers for AWS, Azure and GCP can materialise truly geo-distributed “ephemeral ranges,” while broker gateways can coordinate resources across institutional boundaries. Third, sequence-mining and graph-embedding techniques hold promise for deriving canonical solution paths and feeding intelligent tutoring systems that deliver context-aware hints. Finally, embedding lightweight reinforcement-learning agents at the edge of the telemetry pipeline could yield self-healing, self-teaching scenarios that escalate or de-escalate adversarial pressure in response to learner affect and performance.

In summary, this dissertation reframes observability from an operational afterthought to a pedagogical catalyst. By marrying a microservice-centric architecture with a unified telemetry substrate and validating both in a live teaching environment, it demonstrates that cyber ranges can evolve from static sandboxes into adaptive, data-driven ecosystems. The contributions laid out herein—conceptual, methodological, and empirical—offer a blueprint for educators, researchers, and practitioners seeking to cultivate cybersecurity expertise at scale, while the identified limitations chart a clear course toward more inclusive, intelligent and ethically grounded future work.

References

- [1] Global Growth Insights, “Cyber range for education market size, share, growth, and industry analysis, by types (cyber range as a service, cyber range solutions, cyber range platform), by applications covered (university, vocational colleges and k12), regional insights and forecast to 2033,” 2025, last accessed: 12 Jun 2025. [Online]. Available: <https://www.globalgrowthinsights.com/market-reports/cyber-range-for-education-market-113323>
- [2] E. Ukwandu, M. A. B. Farah, H. Hindy, D. Brosset, D. Kavallieros, R. Atkinson, C. Tachtatzis, M. Bures, I. Andonovic, and X. Bellekens, “A review of cyber-ranges and test-beds: Current and future trends,” *Sensors*, vol. 20, no. 24, p. 7148, 2020, last accessed: February 9, 2025. [Online]. Available: <https://doi.org/10.3390/s20247148>
- [3] D. Stamatopoulos, M. Katsantonis, P. Fouliras, and I. Mavridis, “Exploring the architectural composition of cyber ranges: A systematic review,” *Future Internet*, vol. 16, no. 7, p. 231, 2024.
- [4] P. Nespoli, M. Albaladejo-González, J. A. Ruipérez-Valiente, and J. Garcia-Alfaro, “Scorpion cyber range: Fully customizable cyberexercises, gamification and learning analytics to train cybersecurity competencies,” *ArXiv*, vol. abs/2401.12594, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:267094840>
- [5] I. Priyadarshini, “Features and architecture of the modern cyber range: A qualitative analysis and survey,” *ResearchGate*, 2018, last accessed: February 9, 2025. [Online]. Available: https://www.researchgate.net/publication/327835952_Features_and_Architecture_of_The_Modern_Cyber_Range_A_Qualitative_Analysis_and_Survey
- [6] V. Kampourakis, V. Gkioulos, and S. Katsikas, “A step-by-step definition of a reference architecture for cyber ranges,” *Journal of Information Security and Applications*, vol. 88, p. 103917, 2025, last accessed: February 9, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214212624002199>
- [7] J. Soldani, D. Tamburri, and W.-J. van den Heuvel, “Challenges in monitoring microservices: A systematic literature review,” *IEEE Transactions on Cloud Computing*, vol. 6, pp. 320–332, 2018.
- [8] M. M. Yamin, B. Katt, and V. Gkioulos, “Cyber ranges and security testbeds: Scenarios, functions, tools and architecture,” *Computers & Security*, vol. 88, p. 101636, 2020, last accessed: February 9, 2025.
- [9] M. N. Katsantonis, A. Manikas, I. Mavridis *et al.*, “Cyber range design framework for cyber security education and training,” *International Journal of Information Security*, vol. 22, pp. 1005–1027, 2023, last accessed: February 9, 2025. [Online]. Available: <https://doi.org/10.1007/s10207-023-00680-4>
- [10] D. Wen, “Automated cyber ranges: Design features, architectures, scenarios and impacts,” Undergraduate Honors Thesis, Brigham Young University, 2021, last

- accessed: February 9, 2025. [Online]. Available: https://scholarsarchive.byu.edu/studentpub_uht/182
- [11] W. Duo, M. Zhou, and A. Abusorrah, “A survey of cyber attacks on cyber physical systems: Recent advances and challenges,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022, last accessed: February 9, 2025.
 - [12] “Understanding cyber ranges: From hype to reality,” <https://ecs-org.eu/ecso-uploads/2022/10/5fdb291cdf5e7.pdf>, 2020, wG5 PAPER, March 2020. Retrieved from <https://ecs-org.eu/ecso-uploads/2022/10/5fdb291cdf5e7.pdf>.
 - [13] N. Chouliaras, G. Kittes, I. Kantzavelou, L. Maglaras, G. Pantziou, and M. A. Ferrag, “Cyber ranges and testbeds for education, training, and research,” *Applied Sciences*, vol. 11, no. 4, p. 1809, 2021, last accessed: February 9, 2025. [Online]. Available: <https://doi.org/10.3390/app11041809>
 - [14] N. I. of Standards and Technology, “The cyber range: A guide,” 2023, last accessed: February 9, 2025. [Online]. Available: https://www.nist.gov/system/files/documents/2023/09/29/The%20Cyber%20Range_A%20Guide.pdf
 - [15] K. E. Balto, M. M. Yamin, A. Shalaginov, and B. Katt, “Hybrid iot cyber range,” *Sensors*, vol. 23, no. 6, 2023, cited by: 5; All Open Access, Gold Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85151217631&doi=10.3390%2fs23063071&partnerID=40&md5=bd0d3e58ac575f5d41c4dc33462c5ef7>
 - [16] T. Cruz and P. Simões, “Down the rabbit hole: Fostering active learning through guided exploration of a scada cyber range,” *Applied Sciences (Switzerland)*, vol. 11, no. 20, 2021, cited by: 13; All Open Access, Gold Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117924942&doi=10.3390%2fapp11209509&partnerID=40&md5=a0100c40ddc221a00b050b6894dccf37>
 - [17] S. Alharbi and M. Farnham, “Federated cyber ranges: Integrating multiple training environments,” in *Proceedings of the 2020 International Conference on Cyber Training and Simulation*. IEEE, 2020, pp. 1–8.
 - [18] O. Nock, J. Starkey, and C. M. Angelopoulos, “Addressing the security gap in iot: Towards an iot cyber range,” *Sensors (Switzerland)*, vol. 20, no. 18, p. 1 – 19, 2020, cited by: 14; All Open Access, Gold Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091487425&doi=10.3390%2fs20185439&partnerID=40&md5=2d4c28b7e5a3fb886b550101f2bac2e3>
 - [19] G. Guerrero, G. Betarte, and J. D. Campo, “Tectonic: An academic cyber range,” 2024, Conference paper, cited by: 0. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-8521139358&doi=10.1109%2fARGENCON62399.2024.10735713&partnerID=40&md5=aee05fdbfa47dob1193df9f1bc857fd2>
 - [20] A. P. Luise, G. Perrone, C. Perrotta, and S. P. Romano, “On-demand deployment and orchestration of cyber ranges in the cloud,” vol.

- 2940, 2021, Conference paper, p. 80 – 91, cited by: 3. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114928328&partnerID=40&md5=801179997fec8e278fc08ebd96affoad>
- [21] N. Oikonomou, N. Mengidis, M. Spanopoulos-Karalexidis, A. Voulgaridis, M. Merialdo, I. Raisr, K. Hanson, P. De La Vallee, T. Tsikrika, S. Vrochidis, and K. Votis, “Echo federated cyber range: Towards next-generation scalable cyber ranges,” 2021, Conference paper, p. 403 – 408, cited by: 13; All Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115708906&doi=10.1109%2fCSR51186.2021.9527985&partnerID=40&md5=1ac9ebod0262c1c5471eb6dae27f9789>
- [22] A. Zanasi, D. Cristofori, and G. Giorgi, “The european commission contribution to cybersecurity through the echo project,” 2021, Conference paper, cited by: 1. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85124686306&doi=10.1109%2fCMI53512.2021.9663786&partnerID=40&md5=e7ded2ec596e0f7ac9a339edfa496dbe>
- [23] X. Yin, M. Chen, and L. Wang, “Standardized cyber range scenario descriptions for interoperability,” *Computers & Security*, vol. 93, p. 101750, 2020.
- [24] E. Ear, J. L. C. Remy, and S. Xu, “Towards automated cyber range design: Characterizing and matching demands to supplies,” 2023, Conference paper, p. 329 – 334, cited by: 1; All Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85171734979&doi=10.1109%2fCSR57506.2023.10224940&partnerID=40&md5=ad6613801e57d8706a3fc39bc8968a6a>
- [25] R.-V. Mahmoud, E. Kidmose, A. Turkmen, O. Pilawka, and J. M. Pedersen, “Defatt - architecture of virtual cyber labs for research and education,” 2021, Conference paper, cited by: 9. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114198342&doi=10.1109%2fCyberSA52016.2021.9478236&partnerID=40&md5=594f1e152e57f37d9dcfia8913foac81>
- [26] T. Stodulka and Z. Martinasek, “Introduction of cyber arena,” 2022, Conference paper, p. 506 – 510, cited by: 2. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85162014391&partnerID=40&md5=362eb2ec8066a2f871d76164582599ad>
- [27] M. Katsantonis, A. Manikas, I. Mavridis, and D. Gritzalis, “Cyber range design framework for cyber security education and training,” *International Journal of Information Security*, vol. 22, no. 4, p. 1005 – 1027, 2023, cited by: 14; All Open Access, Hybrid Gold Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85150226378&doi=10.1007%2fs10207-023-00680-4&partnerID=40&md5=6b9de2e2dfc1a9ab5cddf0f2d9fb27d8>
- [28] I. Stamatopoulos and Others, “Leveraging open-source tools for cyber range monitoring,” *Cybersecurity Journal*, vol. 12, no. 2, pp. 101–115, 2023, last accessed: February 9, 2025.

- [29] V. Soundararajan, “Prometheus: An Open-Source Systems Monitoring and Alerting Toolkit,” *USENIX ;login:*, vol. 41, no. 4, pp. 27–33, 2016.
- [30] Grafana Labs, “Grafana documentation,” <https://grafana.com/docs/>, 2021, documentación oficial, acceso abril 2025.
- [31] J. Presnall and M. Radivojevic, “Aggregating xapi statements for cross-platform learning analytics in military cyber exercises,” *Cyber Training Review*, vol. 6, pp. 45–60, 2018, last accessed: February 9, 2025.
- [32] “Asciinema – record and share your terminal sessions,” <https://asciinema.org>, accessed April 2025.
- [33] K. Korc, “logkeys: A gnu/linux keylogger that works,” <https://github.com/kernc/logkeys>, 2015, accessed April 2025.
- [34] “Experience api (xapi) specification,” <https://www.adlnet.gov/xAPI>, accessed May 2025 (IEEE 9274-1-1:2022).
- [35] “Opentelemetry – a framework for observability,” <https://opentelemetry.io>, 2021, cloud Native Computing Foundation project.
- [36] J. Chen, L. Wang, and X. Zhang, “Unified observability for microservices: A case study,” *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 834–847, 2021.
- [37] S. K. Shandilya, S. Upadhyay, A. Kumar, and A. K. Nagar, “Ai-assisted computer network operations testbed for nature-inspired cyber security based adaptive defense simulation and analysis,” *Future Generation Computer Systems*, vol. 127, p. 297 – 308, 2022, cited by: 38. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116004679&doi=10.1016%2fj.future.2021.09.018&partnerID=40&md5=c26aaffc890bebe36e5abad41831defc>
- [38] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, and J. E. McKenzie, “Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews,” *BMJ*, vol. 372, 2021. [Online]. Available: <https://www.bmj.com/content/372/bmj.n160.full.pdf>
- [39] V. Kampourakis, “Secure infrastructure for cyber-physical ranges,” *Lecture Notes in Business Information Processing*, vol. 476 LNBIP, p. 622 – 631, 2023, cited by: 2. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85163413073&doi=10.1007%2f978-3-031-33080-3_45&partnerID=40&md5=2a1941da6e8ea6d5ddc55d5fe0c9cc23
- [40] R. Tan, Q. Tan, H. Wang, Y. Xie, and P. Zhang, “P-i2prange: An automatic construction architecture for scenarios in i2p ranges,” 2024, Conference paper, cited by: 0. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85205026841&doi=10.1109%2fIJCNN60899.2024.10651444&partnerID=40&md5=903dfid4fd43174ff9af797c256da981>

-
- [41] P. Huff, S. Leiterman, and J. P. Springer, “Cyber arena: An open-source solution for scalable cybersecurity labs in the cloud,” vol. 1, 2023, Conference paper, p. 221 – 227, cited by: 4. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85149812178&doi=10.1145%2f3545945.3569828&partnerID=40&md5=22d40976a9f936e085525dcodebccf2a>
 - [42] J. Doe and Others, “Using augmented reality for learning naval operations,” in *Proceedings of the Naval Training Conference*, 2021, accessed May 2025.
 - [43] G. Sharma, E. Sherif, M. He, and E. Boiten, “Analysis of cyber-attacks for modern digital railway system using cyber range,” 2022, Conference paper, cited by: 3. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85160716153&doi=10.1109%2fIATMSI56455.2022.10119321&partnerID=40&md5=fd106e6c44943f5440bc26d7687fbaeb>
 - [44] M. Pihelgas and M. Kont, “Frankenstack: Real-time cyberattack detection and feedback system for technical cyber exercises,” 2021, Conference paper, p. 396 – 402, cited by: 7. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115682044&doi=10.1109%2fCSR51186.2021.9527923&partnerID=40&md5=cf217817879b3f3fc772049f296ab299>
 - [45] D. Cruz, T. Cruz, V. Pereira, and P. Simões, “Designing a high-fidelity testbed for 5g-based industrial iot,” vol. 2023-June, 2023, Conference paper, p. 151 – 160, cited by: 1. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85167620976&partnerID=40&md5=2c4233ea54dcf524f818ff3fa310edae>
 - [46] X. Low, D. Yang, and D. Yang, “Design and implementation of industrial control cyber range system,” 2022, Conference paper, p. 166 – 170, cited by: 2. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85153684775&doi=10.1109%2fCyberC55534.2022.00034&partnerID=40&md5=e47ebobd98a863fa0051e0c17d722216>
 - [47] A. Z. Sharifi, V. Vanijja, D. Pal, and W. Anantasabkit, “Cyberiot: An initial conceptualization of a web-based cyber range for iot,” 2021, Conference paper, p. 91 – 96, cited by: 5. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85128958885&doi=10.1109%2fComPE53109.2021.9752401&partnerID=40&md5=3867196c3b1388d33c4f7c12c38ffe6d>
 - [48] J. Welch, O. Hasan, B. Burch, N. Vey, J. A. Geddes, and K. Hofstra, “Cyberboss: An approach for control and interoperation of cyber for training,” 2020, Conference paper, cited by: 3. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101966582&partnerID=40&md5=465b3f75458c3544bob46319afd726a9>
 - [49] S. Khan, A. Volpatto, G. Kalra, J. Esteban, T. Pescanoce, S. Caporusso, and M. Siegel, “Cyber range for industrial control systems (cr-ics) for simulating attack scenarios,” vol. 2940, 2021, Conference paper, p. 246 – 259, cited by: 5. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114916526&partnerID=40&md5=ddea13a19e84f9617b3feb4965f31c9>
-

- [50] M. Leitner, M. Frank, W. Hotwagner, G. Langner, O. Maurhart, T. Pahi, L. Reuter, F. Skopik, P. Smith, and M. Warum, “Ait cyber range: Flexible cyber security environment for exercises, training and research,” 2020, Conference paper, cited by: 29; All Open Access, Bronze Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123041569&doi=10.1145%2f3424954.3424959&partnerID=40&md5=b261920934f861e2ac39764d990a814b>
- [51] C. Braghin, S. Cimoto, E. Damiani, F. Frati, L. Mauri, and E. Riccobene, “A model driven approach for cyber security scenarios deployment,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11981 LNCS, p. 107 – 122, 2020, cited by: 15. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081679647&doi=10.1007%2f978-3-030-42051-2_8&partnerID=40&md5=0f0b54d9bd928ac76b07c652bfe8abe5
- [52] A. Papoutsaki, T. S. Schmidt, J. M. Rehg, and J. Huang, “WebGazer: Scalable webcam eye tracking using user interactions,” in *Proc. 25th Int’l Joint Conf. on Artificial Intelligence (IJCAI)*, 2016, pp. 3839–3845.
- [53] W. B. HCI, “Democratizing webcam eye tracking on the browser,” <http://webgazer.cs.brown.edu>, accessed April 2025.