A Survey of Cyber Range Training Exercise Scenario Description, Generation, and Execution

Aayush Garg, Abdelwahab Boualouache, Adnan Imeri, and Uwe Roth

***Abstract*—Cyber ranges enable hands-on cybersecurity training. Yet many exercises remain static and labor-intensive, with limited use of current Cyber Threat Intelligence (CTI). This survey takes an end-to-end, three-layer view of the problem as (i) Cyber Range Training Exercise (CyRaTrEx) Scenario Description Languages (SDLs) that formally specify topologies, narratives, and objectives, (ii) CTI-driven scenario generation pipelines that transform threat data into executable content, and (iii) Scenario Execution Platforms that instantiate and operate scenarios. Following a PRISMA methodology, we surveyed 107 publications published between 2010 and 2025, and organized the space into a unified taxonomy. We conduct a comparative evaluation spanning across layers (primarily, SDL, CTI pipeline, execution platform, and cross-layer). This research identified 7 key challenges as (C1) Formal Semantics and Verification, (C2) Behavioural Fidelity, (C3) Trustworthiness of CTI Data, (C4) Continual Scenario Evolution, (C5) API-Driven Scenario Ingestion, (C6) Scalable Telemetry and Observability, and (C7) Reproducibility and Benchmarking, highlighting where the current solutions succeed and where they fall short. Our main contributions are (1) a holistic survey unifying SDLs, CTI pipelines, and cyber range platforms, (2) a taxonomy of representative solutions, (3) a cross-layer gap analysis exposing shortfalls in interoperability, automation, and realism, and (4) a research roadmap toward scalable, realistic, and cyber threat-informed training.**

***Index Terms*—Benchmarking; Cyber ranges; Cyber threat intelligence; Cybersecurity training; Interoperability; Reproducibility; Scenario description languages; Scenario execution; Scenario generation; Telemetry.**

I. Introduction

Cyber ranges are controlled, interactive environments that replicate real-world networks and cyber threats, enabling realistic training exercises [1]. By blending virtual and physical components, a cyber range allows cybersecurity teams to practice defense, incident response, and forensic techniques in a safe setting that closely mirrors modern attack scenarios [2]. Such platforms have gained prominence amid a growing cyber skills gap [3] and increasingly sophisticated threats [4]. In fact, cyber ranges are seen as a key tool to **bridge the cybersecurity training gap** by combining hands-on experience with realistic scenarios and competitions [5]. Open platforms, such as Facebook's CTF framework [6] and the CTFd system [7], have been used to host such exercises. NATO's large-scale exercises, including Cyber Coalition [8], Locked Shields [9], and Crossed Swords [10], further demonstrate the value of realistic range environments in strengthening cyber defense readiness. Industry sources also consider such hands-on cyber range-based training effective in preparing teams for real-world threats [11]. Notably, as early as 2009, experts highlighted the need for such advanced cyber test environments [12], where early work began simulating cyber attacks in lab networks to study worms and other threats [13]. However, **designing and managing effective Cyber Range Training Exercise (*CyRaTrEx*) scenarios** remains challenging, as expert instructors must manually script complex cyber attack narratives, and keeping these scenarios up-to-date with evolving threats is labor-intensive [14], [15].

Over the past decade, researchers and practitioners have sought to automate and standardize aspects of CyRaTrEx scenario creation. One major effort is the development of **CyRaTrEx Scenario Description Languages (*SDLs*)**, domain-specific languages to formally specify the *"who, what, when, and where"* of a training exercise. For example, *Russo et al.* [16] defined a Cyber Range Automated Construction Kit *(CRACK)* SDL with built-in CyRaTrEx scenario verification. Several other SDLs have emerged (YAML/JSON-based or custom XML schemas) to capture network topologies, attacker actions, and defensive measures in a reusable, declarative format (detailed in section IV-A). A second significant effort is the **integration of Cyber Threat Intelligence (*CTI*)**. As adversary tactics change rapidly, there is an interest in leveraging CTI feeds, such as *MITRE ATT&CK* [17] and malware indicators, among others, to generate or adapt scenarios automatically. Recent works introduce CTI-driven CyRaTrEx scenario generation pipelines that transform real-world threat data into simulated incidents [18], [19]. Moreover, a wide array of **CyRaTrEx scenario execution platforms (*cyber ranges*)** has been developed, ranging from academic testbeds to commercial training services (spanning open-source frameworks and vendor-operated platforms [20]–[26]), each providing different levels of automation, fidelity, and scalability. These platforms vary in virtualization support (e.g., virtual machines vs. containers), telemetry instrumentation (e.g., host logs vs. full packet capture), and integration capabilities (e.g., closed systems vs. open APIs for scenario injection).

Despite this progress, early surveys, such as *Davis and Magrath*'s 2013 review of 30 cyber ranges [27], focused on classifying cyber range infrastructures (e.g., simulation vs. emulation, etc.) and documented early testbeds such as the DETER lab for cybersecurity experimentation [28], among others. These surveys observed that only a minority of the cyber ranges featured any automation [29]. More recent studies have examined modern ranges through case studies or interviews [30]. While a recent review of cyber range taxonomies

A. Garg, A. Boualouache, A. Imeri, and U. Roth are with the Luxembourg Institute of Science and Technology (LIST), Luxembourg. Emails: aayush.garg@list.lu; abdelwahab.boualouache@list.lu; adnan.imeri@list.lu; uwe.roth@list.lu;

highlights functional trends in cyber range design [31], none have systematically analyzed the CyRaTrEx scenario content and CTI-driven automation across a broad set of platforms. In other words, we lack a unified view of how scenario description languages, threat intelligence pipelines, and execution platforms intersect to enable end-to-end *intelligence-led* cyber exercises. *This survey aims to bridge this gap.*

We conducted a comprehensive literature review of CyRaTrEx scenario frameworks published between 2010 and 2025, following PRISMA [32] methodology (detailed in section III). From an initial *590* studies, we identified **107 relevant studies** spanning distinct SDL proposals, CTI-driven scenario generators, and execution platforms (cyber ranges). Building on these sources, we organize the domain into a **three-layer** taxonomy (detailed in section IV) covering (i) *Scenario Description Languages (SDLs)*, (ii) *CTI-driven CyRaTrEx scenario generation pipelines*, and (iii) *execution platforms (cyber ranges)*. Also, we perform a comparative evaluation (section V) of the surveyed artefacts along key dimensions like automation capability, cyber attack coverage, and instrumentation fidelity. Drawing on these results, we identify **7 open research challenges** (section VI), i.e., the need for (C1) *Formal Semantics and Verification*, (C2) *Behavioural Fidelity*, (C3) *Trustworthiness of CTI Data*, (C4) *Continual Scenario Evolution*, (C5) *API-Driven Scenario Ingestion*, (C6) *Scalable Telemetry and Observability*, and (C7) *Reproducibility and Benchmarking*, that must be addressed to achieve fully integrated and threat intelligence–driven training. Moreover, we outline avenues for future research (detailed in section VII), building on our identified open challenges.

Thus, our contributions are (1) a comprehensive survey unifying SDLs, CTI pipelines, and cyber range platforms for cybersecurity training, (2) a taxonomy covering 107 studies, enabling rigorous cross-evaluation, (3) empirical insights into the current capabilities and gaps (e.g. prevalence of declarative YAML SDLs, automation levels of containerized vs. VM-based ranges), and lastly (4) future research directions to advance the state of the art towards achieving scalable, realistic, and cyber threat-informed training. The remainder of this survey is organized as section III detailing our survey methodology, section IV introducing the three-layer taxonomy with illustrative examples, section V presenting the comparative evaluation results, section VI discussing the open research challenges, section VII providing future research directions, and finally, section VIII concluding the survey.

II. Related Work and Comparative Positioning

Several works have systematically reviewed aspects of cyber ranges, each with a different focus. For instance, an early and often-cited baseline is the review by ***Davis and Magrath*** [27]. They classified approximately thirty cyber ranges and testbeds by their *underlying infrastructure* (e.g., simulation vs. emulation) and highlighted that only *a few platforms exhibited notable automation* at that time. Their review did not address formal scenario design or the use of cyber threat intelligence *(CTI)* to shape exercise content. Later, ***Yamin et al.*** [29] provided a systematic literature review of cyber ranges and security testbeds and developed a taxonomy of capabilities, including dimensions such as *scenarios, monitoring, roles, and tools*. Their focus was on *architectures and platform features*, where CTI-driven scenario design was outside their scope. ***Ukwandu at al.*** [33] broadens the scope by comparing cyber ranges and testbeds across *domains, users, implementation methods, and applications*, and proposes a multi-dimensional taxonomy to distinguish information technology *(IT)*-focused vs. operational technology *(OT)*-focused environments. ***Chouliaras et al.*** [30] survey *ten cyber range platforms* via structured interviews, uncovering the essential components, topologies, and tools used to create and operate modern ranges, though many commercial and military implementations remain opaque. ***Stamatopoulos et al.*** [34] take an architecture-centric view, identifying key attributes of cyber range infrastructure and highlighting challenges in *cost, automation, and federation* that must be addressed in future designs. More recently, ***Steininger et al.*** [35] focus on *accessibility* of training where they develop a cyber range feature ontology mapping aspects like *platform access and content delivery*, and identify three key trends in practice, i.e., an increasing emphasis on realistic cyber-physical scenarios, widespread automation of range operations, and a reliance on open-source tools and orchestration software. ***Lillemets et al.*** [31] analyze existing cyber range taxonomies, proposing a seven-dimension framework (*Scenario, Environment, Teaming, Learning, Monitoring, Management, Technology*) to capture functional and technical capabilities.

Table I compares these prior surveys along multiple criteria, primarily scope, methodology, and focus. As shown, previous surveys have examined cyber range taxonomies, architectures, component tools, and accessibility. However, a **gap** remains in unifying scenario specification, threat intelligence-informed scenario generation, and scenario execution evaluation. **Our** survey fills this gap by focusing on **cyber range training exercises**, specifically the chain from formal training exercise scenario description languages through cyber threat intelligence-driven scenario generation to execution platforms. We offer a holistic, *three-layer framework* including Scenario Description Languages *(SDL)*, Cyber Threat Intelligence *(CTI)*- driven scenario generation pipelines, and scenario execution cyber range platforms. Moreover, we perform a structured comparison of how well each layer supports automation and realistic training. This positions our survey as a bridge between cyber range design and its use, focusing on the realism and automation needs of modern training exercises.

III. Survey Methodology

A. Survey Design and Scope

Our survey aims to provide a comprehensive overview of the emerging ecosystem of CyRaTrEx scenario design, generation, and execution. In particular, it covers three interrelated aspects essential for intelligence-driven cyber defense training, primarily, (i) scenario description languages (SDLs) for formally specifying cyber exercise content, (ii) cyber threat intelligence (CTI) pipelines that generate or adapt scenarios from real-world threat data, and (iii) cyber range execution platforms

TABLE I: Comparative Analysis of Related Surveys

Survey	Year	Scope	Methodology	Key Contributions (Focus)
Davis & Magrath [27]	2013	Early cyber ranges and testbeds	Literature review (cyber range classification)	Classified cyber ranges by infrastructure (simulation vs. emulation). Noted that a few platforms had automation. Formal scenario design and CTI were not addressed.
Yamin et al. [29]	2020	Cyber ranges & security testbeds	Systematic literature review	Developed a taxonomy of cyber range capabilities with six dimensions (scenario, monitoring, etc.). Focused on cyber range architecture and scenarios (tools, roles). CTI was not addressed.
Ukwandu et al. [33]	2020	Cyber ranges and OT testbeds	Systematic review	Broad review covering cyber ranges and OT testbeds, segmented by type, technology, threat scenarios, etc. Introduced taxonomies and noted a diminishing gap between them.
Chouliaras et al. [30]	2021	Educational/training cyber ranges	Survey of 10 systems and interviews	Combined a systematic study of 10 ranges with structured interviews of developers. Identified key components, tools, and practices for cyber range design and operation.
Stamatopoulos et al. [34]	2024	Cyber range architectures	PRISMA systematic review	Explored the architectural composition of cyber range (infrastructure layers). Found limited prior work on component interoperability. Highlighted gaps in design and administration.
Steininger et al. [35]	2025	Cyber range accessibility, technology stack, automation	Literature review and practitioner survey	Investigated accessibility of training (platform access, content, deployment). Developed a feature ontology mapping cyber range characteristics (technology stack, automation, onboarding).
Lillemets et al. [31]	2025	Cyber range taxonomies	Systematic literature review	Surveyed existing cyber range taxonomies and proposed a new one decoupling functional features from underlying technology. Added a dedicated dimension highlighting AI and federation trends.
Our Survey	2025	Training exercise scenario description, cyber threat intelligence-informed scenario generation, and scenario execution	PRISMA systematic review	Presents a holistic end-to-end view of cyber range training exercise scenarios. Links scenario description languages *(SDLs)*, cyber threat intelligence *(CTI)* -driven scenario generation, and scenario execution platforms. Provides a three-layer taxonomy, cross-layer comparison, and identifies key gaps across dimensions, including automation, realism, and interoperability.

that deploy and orchestrate these scenarios in realistic environments. By unifying these aspects in this survey, we aim to illustrate how advances in scenario specification, threat-intel utilization, and platform capabilities collectively enable more effective and automated cybersecurity exercises.

The scope of our review encompasses peer-reviewed research publications (2010–2025) that introduce, extend, or evaluate solutions in any of the above categories. We intentionally focus on works that contribute to the automation or formalization of CyRaTrEx scenarios, such as proposing a new SDL, integrating CTI sources into scenario generation, or developing platforms with novel scenario-execution features. In contrast, we exclude literature centered on peripheral topics, such as general cybersecurity training methods or proprietary range offerings lacking technical detail, since these fall outside our technology-oriented focus. By establishing the survey in this way, we ensure a targeted synthesis of the innovations that drive modern cyber range exercises, laying the groundwork for the taxonomy and analysis presented in subsequent sections.

B. Search and Selection Process

We conducted a comprehensive search to identify relevant literature on the topic of cyber range design, scenario specification, and cybersecurity simulation. The search was performed across multiple electronic databases, including IEEE Xplore, ACM Digital Library, Scopus, Web of Science, and the arXiv preprint server. We employed search queries combining key terms of interest, for example:

- *"cyber range"*,
- *"scenario description language"*,
- *"cyber threat intelligence integration"*,
- *"cybersecurity simulation"*,
- and related keywords (e.g., "cyber exercise", "simulation framework", "testbed").

The initial database searches returned on the order of several thousand records. All retrieved items were aggregated, and duplicate entries (exact or near-duplicate titles) were removed, leaving a subset of unique records for further screening. The screening and selection followed a PRISMA-like protocol [32] where in the first phase, titles and abstracts of all unique records were examined to exclude publications outside the scope (e.g., non-technical articles or unrelated domains). In the second phase, we performed a full-text review of the remaining papers and excluded any studies not meeting our inclusion criteria.

To ensure transparency, the filtering process is summarized below:

1) **Identification:** Initial search yielded approximately *590* records from all sources.
2) **Deduplication:** Removing duplicate records resulted in *478* unique records.

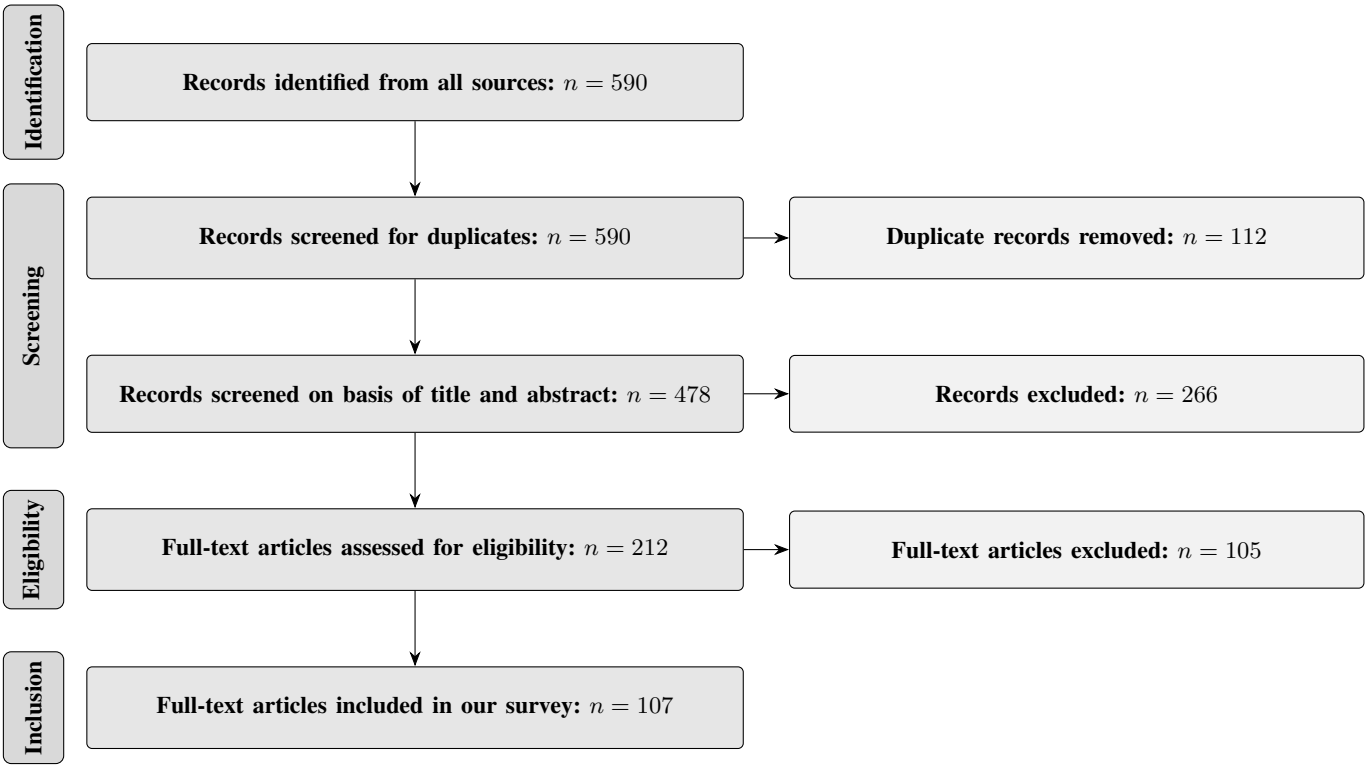


Fig. 1: PRISMA flow diagram summarizing our search and selection process.

3) **Screening:** *478* titles/abstracts were screened, yielding *212* potentially relevant papers.
4) **Eligibility:** Full texts of those *212* papers were assessed against our criteria (*Section III-C*), with *107* papers deemed relevant.
5) **Inclusion:** The final survey includes *107* publications that met all inclusion criteria.

Figure 1 illustrates the PRISMA flow of our search and selection, from identification and deduplication through screening, eligibility, and inclusion (assessed against the criteria in the following section *III-C*).

C. Inclusion and Exclusion Criteria

We applied the following **Inclusion Criteria** (IC_n) and **Exclusion Criteria** (EC_n) when selecting publications for our survey:

IC_1 Focuses on one or more of our primary topics, i.e, scenario description languages (SDLs), CTI-driven scenario generation, or cyber range execution platforms.
IC_2 Contains considerable technical content (e.g., detailed methods, architectures, or evaluations) and appears in a peer-reviewed venue or as a credible technical report.
IC_3 Is written in English and published between 2010 and 2025 (inclusive).

We excluded items that:

EC_1 Are opinion pieces, editorials, blog posts, or short articles without original technical content.
EC_2 Focus exclusively on training, education, or gamification aspects without contributing to SDL/CTI design or infrastructure development.
EC_3 Are duplicate publications or secondary versions of other included works.
EC_4 Lack an accessible full text for review.

In addition to peer-reviewed sources, we also considered selected grey literature (e.g., open-source documentation and technical white papers) from reputable organizations, provided it contained essential methodological or implementation details not found in the peer-reviewed literature.

D. Classification Criteria

To systematically organize the 107 selected papers, we devised a multi-dimensional classification scheme. Each paper was categorized into one or more of three thematic domains, primarily *Scenario Description Languages* (SDLs), *CTI-driven Scenario Generation Pipelines*, and *Cyber Range Execution Platforms*. Notably, some works address multiple aspects of scenario generation. Classification was performed independently by two reviewers, with conflicts resolved by discussion and consensus. The review process followed standard systematic literature review practices.

Specifically, our classification scheme considers the following sub-dimensions within each domain:

1) **Scenario Description Languages (SDLs):** SDLs are characterized by their syntax, validation support, and reusability. In particular:
   a) ***Syntax type***, e.g., YAML vs. XML, reflecting the data-serialization format used to define scenarios.
   b) ***Validation support***, indicating whether the language provides schema definitions or tool-based checking to ensure scenario correctness.

c) ***Reusability***, i.e., the support for modular or template-based scenario components that can be reused or shared.

2) **CTI-driven Scenario Generation Pipelines:** The pipelines are characterized by their threat intelligence sources and automation. The key dimensions include:
   a) ***Cyber Threat source type***, describing the origin of threat information (e.g., open-source feeds, proprietary CTI, or simulated data).
   b) ***Automation level***, indicating whether scenario generation is manual, semi-automated, or fully automated.
   c) ***CTI framework (and platform) mapping***, denoting the degree to which the pipeline aligns generated CyRaTrEx scenarios with tactics and techniques provided by the CTI frameworks (e.g., *MITRE ATT&CK* [17]) and platforms (e.g., *MISP* [36]).
3) **Cyber Range Execution Platforms:** Platforms are classified by their infrastructure and instrumentation characteristics as follows:
   a) ***Virtualization support***, e.g., hypervisor-based vs. software-defined (e.g., container or cloud) infrastructure.
   b) ***Instrumentation fidelity***, indicating the level of realism or monitoring granularity (e.g., full system emulation vs. lightweight simulation).
   c) ***Openness***, i.e., whether the platform implementation is open source or proprietary.

IV. The Cyber Range Taxonomy

The key elements of a threat-informed cyber range can be organized hierarchically into scenario description languages, generation methods, threat-intelligence sources, and platform types. For example, Scenario Description Languages (SDLs) fall into two main families: **general-purpose data formats** (e.g. *JSON [37], YAML [38], XML [39]*) and **specialized domain-specific languages** (e.g. *CRACK SDL [16], VSDL [40], CST-SDL* [41]). Scenario Generation Methods includes *manual authoring, replay-based (record-and-replay) techniques, AI/ML-driven generation,* and *hybrid approaches* that combine these strategies (e.g., feature-based model generation [42]). Cyber Threat Intelligence (CTI) typically leverages structured frameworks like *MITRE ATT&CK* [17] and sharing platforms like *MISP* [36] to inform realistic scenarios. Finally, CyRaTrEx scenario execution platforms are often categorized as **open-source/academic** and **commercial/government** solutions. We illustrate this taxonomy in **Figure 2**, and detail every aforementioned component in a subsequent sub-section, respectively (sub-sections IV-A–IV-F).

A. Scenario Description Languages (SDLs)

Cyber Range Training Exercise Scenario Description Languages (*CyRaTrEx SDLs/SDLs*) provide a structured, machine-readable specification of all elements of a cyber range training scenario (e.g., network topology, host configurations, attacker/defender roles, and exercise events) to enable automated deployment and execution of training exercises. SDLs enable training environments to be modeled abstractly and then automatically instantiated on a cyber range [41]. In practice, SDLs fall into two broad categories, as detailed below.

1) **General-purpose formats:** Many existing ranges use familiar data-serialization formats such as **JSON** [37], **YAML** [38], or **XML** [39] with custom schema definitions or templates to describe scenario elements. These formats are human-readable and leverage existing tooling. In this approach, a scenario file declaratively lists components (VMs, networks, links, etc.) and configuration parameters, which are then consumed by deployment tools (e.g., translating a YAML description into *OpenStack Heat* [43] templates and *Ansible* [44] playbooks) [40]. Standard JSON/YAML/XML schemas provide basic validation (syntax checking, required fields, simple type checking) but generally capture only infrastructure and static parameters. Higher-level aspects (e.g. narrative events, objectives, scoring) are typically handled outside the schema or in ad hoc platform-specific fields. As a result, each platform's schema tends to be isolated, where the scenarios written for one system are not directly portable to another without translation.
2) **Domain-specific SDLs:** To overcome these limitations, specialized scenario description languages have been proposed that embed rich semantics and built-in validation. For example, the **Topology and Orchestration Specification for Cloud Applications (*TOSCA*)** [45] is an *OASIS* standard to describe cloud-based applications in an interoperable manner. The **Cyber Range Automated Construction Kit *(CRACK)* SDL** [16] (developed under the *EU SPARTA* project) extends the TOSCA standard using a YAML-based syntax. CRACK SDL can describe not only cloud infrastructure (compute nodes, networks, subnets, etc.) but also cyber-specific elements, such as vulnerabilities, attacker/defender teams, and training objectives. The TOSCA foundation provides inheritance and templates for composing components, and the CRACK toolchain translates the SDL into deployable artifacts and verifies the scenario's logical consistency (via a Datalog-based check) before execution [16]. Similarly, the **Collaborative Security Training SDL (*CST-SDL*)** [41] provides a custom graph-based SDL for multi-trainee exercises, allowing explicit definition of roles, goals, and solution paths. The **Virtual Scenario Description Language (*VSDL*)** [40] is another domain-specific SDL for high-level infrastructure features. It uses constraint-based semantics and **Satisfiability Modulo Theories (*SMT*)** [46] to automatically generate concrete scenarios from a compact specification. Similarly, *Costa* and *Kuusijarvi* have also proposed a programmatic topology description language for cyber ranges [47]. These specialized SDLs typically adopt a declarative syntax and enhance it with formal validation and domain concepts.

Regarding the **comparison of SDLs**, these can be compared along several key dimensions, as we detail below.

1) **Syntax format:** The syntax of SDL ranges from simple structured schemas to full-fledged SDLs. General formats

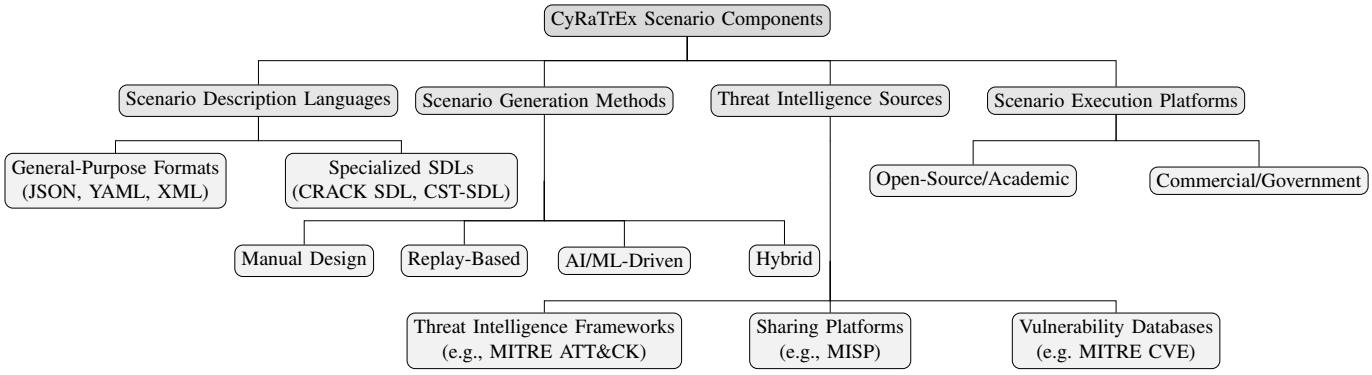


Fig. 2: Taxonomy of the key CyRaTrEx scenario components, primarily scenario description languages, scenario generation methods, threat-intelligence sources, and scenario execution platform.

use standard notation (e.g., JSON, YAML, or XML) that is widely supported but whose schema is defined case-by-case by each platform. Specialized SDLs often reuse or extend infrastructure-as-code *(IoC)* standards. For instance, CRACK uses a TOSCA-based YAML syntax, whereas CST-SDL and VSDL have custom textual SDLs tailored to training scenarios. In all cases, the syntax is declarative where authors specify what scenario elements should exist (nodes, networks, exploits, etc.) rather than how to execute them [16], [40].

2) **Validation capabilities:** Generic SDLs rely on conventional schema validation (e.g. JSON Schema or XML Schema) to catch syntactic errors and enforce basic constraints. By contrast, domain-specific SDLs can incorporate richer verification. For instance, CRACK SDL translates the scenario model into a formal Datalog representation to check that the intended attack paths and objectives are consistent before deployment [16]. VSDL uses an SMT solver to ensure that a high-level feature specification can be satisfied by some concrete infrastructure [40]. Such formal checks help detect logical issues (e.g. unreachable goals or conflicting properties) that simple schemas cannot capture.
3) **Modularity and Reusability:** Reusable SDLs allow scenario components (such as network segments or attack phases) to be defined once and reused across scenarios. Basic JSON/YAML schemas may allow inclusion of separate files (e.g., separate topology and configuration sections), but have limited support for abstraction. In contrast, specialized SDLs often provide explicit mechanisms, e.g., CRACK SDL leverages TOSCA's type derivation, enabling users to define new component types by extending base types [16]. VSDL allows scenarios to be composed from smaller feature blocks, and CST-SDL's graph model supports reusing subgraphs of scenario goals. These modular constructs (in CRACK SDL, VSDL, CST-SDL) promote parameterized scenarios and libraries of reusable scenario pieces [16], [40], [41].

The general-purpose JSON/YAML/XML schemas are widely used due to their simplicity and tool support, but they typically capture only low level infrastructure and require platform-specific extensions for *pedagogical* details. Domain-specific SDLs (e.g. CRACK SDL, CST-SDL, VSDL) offer richer semantics and automation: e.g., CRACK's TOSCA-based language enables end-to-end generation, deployment, and testing of scenarios [16], and VSDL's constraint-based SDL supports automated scenario generation and verification [40]. However, these specialized languages are more complex and have seen less widespread adoption. The trade-offs among syntax expressiveness, validation strength, and modularity in each SDL reflect ongoing challenges in the literature where achieving interoperability and reusability requires common standards and tooling across these approaches [16], [40].

Building on the above observations, we extract the **four fundamental requirements** that an ideal Scenario Description Language (SDL) should fulfill to support effective, intelligence-driven cyber ranges. These core requirements, i.e, **interoperability**, **automation**, **semantic richness**, and **reusability**, have been highlighted in recent analyses as critical to overcoming current SDL limitations. We briefly define each requirement below in the context of Cyber Range Training Excercise (*CyRaTrEx*) scenarios:

1) **Interoperability:** SDL interoperability is the ability to use a scenario specification across different cyber range platforms and tools without requiring tedious manual translation. In practice, this means adopting common standards or exchange formats so that scenarios written for one system can be parsed and executed on another. High interoperability ensures that scenario content is portable and shareable, which in turn fosters collaboration and avoids *isolated* SDL ecosystems. Achieving this requires standardized schemas and mappings (e.g., a common core SDL or adapters between formats) so that heterogeneous platforms can understand the same scenario description. Recent efforts like the ***ECHO project's unified SDL*** [48] and other proposals underscore the community's push toward greater SDL interoperability.
2) **Automation:** Automation in the context of SDLs refers to supporting the automatic generation, deployment, and execution of scenarios with minimal human intervention. An SDL should integrate with orchestration tools and cyber range pipelines to allow scenarios to be instantiated and run end-to-end by machines. High automation means that given an SDL file, the cyber range can automatically

set up the network topology, configure systems, launch scripted events, and even adapt scenario parameters on the fly. This capability is crucial for scalability as it enables dynamic scenario synthesis from threat intelligence feeds and reduces the manual scripting burden on exercise designers.

3) **Semantic Richness:** Semantic richness (*or expressiveness*) denotes the ability of an SDL to capture a high-level scenario, i.e., in addition to describing the static infrastructure, it should also describe the roles, behaviors, and narrative context. A semantically rich SDL can encode complex concepts such as attacker tactics, defender actions, training objectives, and event timelines in a machine-readable format. This goes beyond simple configuration, meaning that the language has constructs for scenario logic (e.g., attack steps, dependencies, win/lose conditions) and can represent abstract concepts (such as *privilege escalation occurring via a specific exploit*) rather than only listing VMs and IP addresses. Rich semantics enable advanced features such as formal verification of scenario properties (e.g., checking that all attack paths are reachable) and alignment with cyber threat intelligence frameworks (i.e., mapping scenario elements to frameworks such as MITRE ATT&CK tactics, etc.). Semantic richness ensures the SDL can describe ***what happens in a scenario at a conceptual level***, which is key for realistic and meaningful training exercises.
4) **Reusability:** Reusability is the capacity to create modular scenario components and templates that can be reused or adapted across multiple exercises. A reusable SDL lets authors define scenario building blocks (e.g., a network segment, an attack vector, or a role-playing element) once and incorporate them into many scenarios, possibly with different parameters. This modular design approach saves effort and promotes consistency, as common elements do not need to be rewritten for each new scenario. Reusability also ties closely to interoperability, as scenarios with standardized and modular definitions are easier to share via libraries or repositories and can be imported into different platforms. Features that support reusability include inheritance or templating mechanisms in the language (such as CRACK SDL's TOSCA-based type derivation) and the ability to parameterize scenarios. Ultimately, emphasizing reusability in SDL design helps build an ecosystem of ready-made scenario parts and encourages continuous improvement of scenarios over time, rather than ad hoc one-off creations.

B. Scenario Generation Methods

CyRaTrEx scenarios can be produced through a spectrum of methods, ranging from fully manual design to fully automated generation. We distinguish four broad scenario generation approaches, primarily **Manual Design**, **Replay-Based Generation**, **AI/ML-Driven Generation**, and **Hybrid Approaches**, each with distinct workflows, strengths, limitations, and pedagogical implications. We detail each approach below.

1) **Manual Design:** In the traditional method, human experts hand-craft the entire scenario environment, narrative, and event timeline. This approach offers **maximal pedagogical alignment** as instructors can tailor every detail to specific learning objectives, ensuring scenarios directly target desired skills. Manually built scenarios also afford high fidelity if designers invest effort (e.g., realistic network topologies and attack traces). However, the downsides are significant, e.g., manual development is **labor-intensive and not easily scalable**, often taking weeks of effort per scenario. Keeping scenarios up-to-date with evolving threats requires continual human updates. Moreover, purely manual scenarios tend to be **static**, as the sequence of events is pre-scripted and generally cannot adapt to unexpected trainee actions. Certain tools (e.g., ***SecGen*** [49]) have been introduced to generate vulnerable scenario VMs for training labs automatically; however, purely manual scenarios still dominate. While trainees benefit from a carefully controlled exercise, they may miss out on dealing with unplanned adversary behaviors. In other words, manual authoring/design of CyRaTrEx scenarios provides **high control and customization at the cost of low adaptability and high overhead**. Today, many cyber training programs still rely on manual design (e.g., using YAML/JSON templates in platforms like KYPO [50], [51]) due to simplicity; however, this limits the number and variety of available CyRaTrEx scenarios that can be generated.
2) **Replay-Based Generation:** This approach builds scenarios by **reproducing recorded cyber incidents or prior exercise data** instead of scripting from scratch. For example, captured network traffic (i.e., **Packet capture (*PCAP*)** files), system logs, or sequences of attacker commands from a real incident can be injected into the range environment to recreate the incident faithfully. A concrete example of this approach is the study by *Hussain et al.* [52] where they replay the real attack traffic captured by the LANDER project, on a testbed to study an Internet-scale incident within hours of its occurrence. The strength of replay-based scenarios is **realism through authenticity**, since the content comes from real attacks or well-defined test runs, the training exercise includes all the subtle characteristics of real-world events (timing variances, nuanced attacker behaviors, background noise, etc.). Trainees get to observe and respond to real attack patterns, which is invaluable for learning to recognize real incidents. Replays also ensure consistency; every run replays the same sequence, which is useful for standardized evaluation across trainees. However, there are certain **limitations** to replay-based scenario generation. Replay-driven scenarios are typically **inflexible and non-interactive**. The attack timeline unfolds in a predetermined way regardless of defender actions, e.g., if a trainee deviates or neutralizes the threat early, the scripted replay might continue unrealistically. This lack of adaptability means replay scenarios are best for training detection and response to known attack patterns or for after-action analysis, rather than for free-form adversarial exercises. Another drawback is **dependence on available recordings** as scenarios are limited to the incidents

for which the captures are available. Novel or evolving threats cannot be simulated via replay until data from such incidents is obtained. Despite these caveats, replay-based generation is used in practice (e.g., replays of infamous attacks or malware outbreaks) to provide high-fidelity drills where studies have shown that reusing real attack traces yields more realistic training than fully synthetic simulations [52], [53].

3) **AI/ML-Driven Generation:** In AI-driven scenario generation, artificial intelligence (such as machine learning models or planning algorithms) automates the creation and orchestration of scenario events. These techniques promise **dynamic and adaptive exercises** that can respond to trainee actions in real-time. One instantiation is using **Reinforcement Learning (*RL*)** agents as virtual attackers, e.g., *Microsoft's CyberBattleSim* framework [54] employs an RL agent to autonomously infiltrate a simulated network, generating a realistic, evolving attack sequence as the scenario. Such an AI red team will probe defenses and pivot through networks in unscripted ways, forcing trainees to react to an intelligent adversary. Another direction is using **generative AI models** (e.g., Large Language Models a.k.a. *LLMs*) to create scenario content, given a high-level prompt, an AI might generate a detailed attack narrative, network configuration, or even malicious artifacts, and can **adapt its behavior in real time** based on the defender's actions. AI planning systems have likewise been explored to assemble attack plans (e.g., using attack graphs or logic-based planners to choose exploit sequences). The key advantage of AI-driven methods is **adaptability** as no two runs need be the same, and difficulty can be adjusted in real-time, providing a **personalized, game-like learning experience** for participants. Trainees must think strategically and cannot rely on scripted patterns, which can improve higher-order skills like improvisation and resilience. AI-driven scenarios also address scalability, as once an AI model or pipeline is developed, it can generate countless unique scenarios or simultaneous exercises with minimal extra human effort. However, **challenges** accompany these benefits. AI-generated scenarios introduce **complexity and unpredictability**, instructors hand over some control to the machine, which may result in unexpected behaviors. There is a risk an autonomous agent might exploit the environment in unanticipated ways that do not align with training goals (e.g., finding a shortcut that avoids the intended learning point). Ensuring the AI's actions remain *pedagogically relevant* thus requires careful design and possibly human oversight or constraints. Additionally, developing and validating these AI systems is non-trivial as it demands expertise in **Machine Learning (*ML*)**, substantial scenario data or knowledge bases, and thorough testing to avoid erratic outputs. Thus, AI/ML-driven generation can offer *unparalleled dynamism and realism* for cyber exercises, at the cost of *higher complexity and the need for robust control mechanisms* to keep the training on track.

4) **Hybrid Approaches:** Most state-of-the-art cyber ranges incline toward *hybrid scenario generation* that **combines manual and automated techniques**. The goal is to get the **best of both worlds**, i.e., human instructors set high-level objectives and ensure alignment with learning goals, while automation handles the tedious, complex, or adaptive aspects of scenario creation. There are multiple forms of hybridization. One common pattern is **template- or SDL-driven generation** where an expert designs the scenario in abstract terms (e.g., specifies required network elements, vulnerabilities, and goals in a scenario description language like *VSDL* [40] or *CRACK SDL* [16]), and an engine automatically instantiates a concrete scenario that meets those specifications. This approach leverages human insight for the outline and AI (or automation) for the details. For instance, the CRACK framework can take a high-level declarative scenario description and generate a full deployment complete with configured attacks, even verifying logical consistency before execution. Another hybrid model is a **module library with randomization** where instructors prepare a library of building blocks (e.g., VM images, attack scripts, background traffic generators), then the system assembles scenarios from these blocks with slight random variations. For instance, an exercise might always involve a ransomware attack on a server, but the platform automatically varies the malware variant or the target system on each run, adding unpredictability while following the instructor's general script.

Another hybrid technique is **human-in-the-loop AI**, where the instructor defines the scenario setup and high-level story, but inserts an AI agent or automated scripts to play certain roles during execution (e.g., an autonomous red-team agent to conduct the attack). Here, the **environment and objectives are human-defined** while the **tactics are AI-driven**, yielding a mix of controlled and adaptive behavior [55]. The strength of these hybrid approaches are that they are **balanced** and pragmatic. They significantly **reduce manual effort** (automation may handle environment provisioning, event scheduling, and data generation) while preserving instructor control over critical aspects (objectives, overall narrative structure). Well-designed hybrids can improve **realism** and variability (via automation) without sacrificing pedagogical intent. They also enhance **scalability**, as instructors can reuse abstract scenario templates and let the system generate many concrete variations. Nevertheless, there are certain limitations of this hybridization as these methods can introduce **complexity** in tool integration, e.g., requiring a robust scenario description language and orchestration engine, and may still not achieve the full autonomy of AI or the full degree of human curation. However, in practice, hybrids have proven to be the *most viable solution* for current cyber ranges, **blending manual insight and automated execution**. Many modern platforms (academic and commercial alike) implement this mix, e.g, Airbus's cyber range [56] uses instructor-defined building blocks combined with automated orchestration, making hybrid scenario generation essential for scalable,

intelligence-informed cyber training.

C. CTI Data Exchange Standards

A foundational aspect of automated **Cyber Threat Intelligence (*CTI*)** integration is the use of standardized formats and protocols for sharing threat information across tools. **Structured Threat Information eXpression (*STIX*)** is a widely adopted language for representing cyber threat intelligence in a structured, machine-readable form [57]. STIX defines a consistent schema for entities such as indicators (IOCs), adversary TTPs, campaigns, and threat actors, enabling disparate systems to exchange rich threat data with common semantics. Complementing STIX is the **Trusted Automated Exchange of Indicator Information (*TAXII*)** protocol, which provides an API-driven mechanism to distribute and synchronize CTI feeds between organizations in real time [58]. TAXII servers allow security platforms to *publish* intelligence collections (e.g., new malware indicators or attack patterns) and for others to *subscribe* and fetch updates automatically, leveraging HTTPS and access controls for secure sharing. Together, STIX and TAXII facilitate an automated CTI exchange pipeline: a CyRaTrEx scenario generator can ingest up-to-date threat reports in STIX format via TAXII services, ensuring that exercise scenarios reflect the latest attacks and indicators without manual data conversion. This standards-based interoperability is crucial for maintaining *realism and relevance* in cyber range operations, as it allows training environments to seamlessly incorporate live threat intelligence feeds used in the real world.

Another key standard driving automated security content sharing is the **Security Content Automation Protocol (*SCAP*)** [59] developed by **National Institute of Standards and Technology (*NIST*)** [60]. SCAP is a framework of interoperable specifications designed to standardize how software vulnerabilities, configuration baselines, and compliance information are represented and exchanged. It bundles common enumerations (such as **Common Vulnerabilities and Exposures (*CVE*)** [61] identifiers for known vulnerabilities) for expressing machine-readable vulnerability definitions and configuration checklists. By using SCAP content, security tools can automatically *assess and report* a system's state against known vulnerabilities and policy benchmarks in a uniform way [59]. In the context of cyber ranges, SCAP enables scenario designers to incorporate up-to-date vulnerability data and compliance profiles into exercise environments. For example, a range scenario can include hosts with specific CVEs and misconfigurations that are described via SCAP datasets, allowing trainees to practice detecting and remediating them with standard tools. Integrating SCAP thus enhances the *fidelity of cyber range operations*, exercises can closely mirror enterprise vulnerability management processes (e.g., running SCAP-compliant scanners or patching workflows), and the results can be automatically evaluated against industry-standard criteria. Overall, the SCAP standard ensures that CyRaTrEx scenarios remain aligned with real-world security configuration practices and can rapidly adapt to new vulnerabilities or policy changes.

D. CTI-Driven Scenario Generation Pipelines

Cyber Threat Intelligence (*CTI*)-driven scenario generation pipelines are automated systems that translate up-to-date cyber threat intelligence into executable training scenarios for cyber ranges. In such pipelines, diverse CTI inputs, such as adversary **Tactics, Techniques, and Procedures (*TTP*)** frameworks, threat feeds, and vulnerability databases, are consumed and processed into structured scenario content. For example, common CTI sources include the *MITRE ATT&CK* [17] framework of adversary tactics and techniques, open threat-sharing platforms like *MISP* [36], and vulnerability repositories (e.g. the *CVE* list [61]), often exchanged in standard *STIX/TAXII* format [57], [58]. By leveraging CTI, the generated scenarios reflect realistic and current threats, helping exercises simulate the latest attacker behaviors.

Architecturally, CTI-driven scenario generation pipelines can be classified into three high-level models, that we detail below.

1) **Rule-Based Mapping:** Static rules or mappings convert CTI elements to scenario events. For instance, a rule might map a reported phishing technique in CTI to a pre-scripted email-injection event in the scenario. Such systems rely on hand-crafted associations between known threat indicators and scenario primitives.
2) **Template-Driven Generation:** Predefined scenario templates are parametrized with CTI data. In this model, templates (e.g. network topologies, attack graphs, or storyline fragments) contain placeholders that are filled using specific CTI details (malware names, IP addresses, exploits, etc.). This allows rapid assembly of scenarios around current threats by swapping in fresh intel into template slots.
3) **Hybrid ML-Supported Generation:** ML and natural language techniques are employed to interpret CTI and generate scenario content dynamically. For instance, an AI-based pipeline might use Named-Entity Recognition to extract threat actors and attack stages from text and then formulate scenario steps accordingly. Frameworks like **AI-assisted Cyber Exercise content generation Framework (*AiCEF*)** [18] integrate ML components with rule/template elements, using an ontology (e.g. the Cyber Exercise Scenario Ontology) to structure the output. Such hybrid pipelines can adapt to novel CTI without requiring exhaustive manual encoding.

A typical CTI-driven scenario generation pipeline flow proceeds in stages. First, the pipeline **collects and ingests raw CTI data** from the chosen sources. Next, it applies **parsing and analysis** (often with **Natural Language Programming (*NLP*)** or pattern matching) to extract relevant artifacts such as identified TTPs, attack indicators, threat actors, or exploited vulnerabilities. These extracted elements are then **correlated and mapped to scenario-building blocks**, e.g., an extracted *ATT&CK* technique might be translated into a corresponding simulation action (sending a crafted email, spawning a malicious process, etc.), and a CVE may determine a simulated software exploit step. Often the pipeline enriches this mapping with contextual details (e.g. deriving plausible network targets

or timing) and organizes the elements according to a scenario structure or template. Finally, the pipeline **assembles the scenario by sequencing events** and generating any needed artifacts (scripts, network configurations, logs), producing a **complete scenario output ready for use** in the cyber range. A few representative examples exist that illustrate these ideas, including the AiCEF [18] system, that automatically processes security news articles using named-entity recognition and clustering to extract threats and then composes structured scenarios based on an internal ontology. *Zacharis et al.* [62] describe an AI-driven pipeline that integrates threat forecasting and scenario generation in a single flow, producing sector-specific exercise content. Similarly, prototype tools have been proposed to ingest CTI feeds or reports and map them through defined rules/ontology to create scripted cyber exercises.

Despite their promise, CTI-driven scenario generation pipelines face several challenges. We deatil below the major 3 challenges.

1) **Semantic Alignment:** CTI is often high-level and unstructured (e.g. narrative reports), whereas scenario content requires precise, low-level actions. Misalignment between the semantics of CTI data and the required simulation steps can lead to mismatches or incoherent scenarios if not carefully managed.
2) **Freshness and Coverage:** CTI feeds can become outdated or incomplete. Pipelines must handle stale data and ensure that scenarios remain relevant to current threats. It can be difficult to guarantee that all emerging threats are captured, especially when intelligence sources lag behind real-world events.
3) **CTI-to-Action Gaps:** Translating abstract threat descriptions into concrete injects involves an abstraction gap. For example, a report may note a generic "credential dumping" technique, but the pipeline must decide how to simulate that (which accounts, what tools, timing, etc.). Bridging this gap often requires additional context or heuristics beyond what raw CTI provides.

These limitations indicate that human oversight or expert tuning is often needed to validate and refine generated scenarios. Nonetheless, CTI-driven pipelines offer a path toward more scalable, current, and relevant cyber exercise content by systematically leveraging real threat intelligence.

In addition to ***ATT&CK*** and ***MISP***, several other CTI frameworks and platforms enrich scenario generation and threat modeling. **Common Weakness Enumeration (*CWE*)** [63] is a community-developed catalog of software and hardware weakness types maintained by *MITRE*. CWE provides a standardized taxonomy of vulnerabilities, helping scenario designers incorporate realistic weakness-based challenges and secure coding exercises into CyRaTrEx scenarios. **MITRE's Cyber Analytics Repository (*CAR*)** is a public knowledge base of behavioral detection analytics built on the ATT&CK framework. Each CAR analytic maps to specific adversary techniques, enabling the development and validation of detection logic in training scenarios (e.g., simulating SIEM alerts and threat hunts). **MITRE Detection, Denial, and Disruption Framework Empowering Network Defense (*D3FEND*)** [64] is a structured knowledge graph of defensive techniques that complements ATT&CK by enumerating countermeasures for known tactics. D3FEND provides guidance on defensive measures (hardening, detection, deception, etc.) to mitigate or respond to simulated attacks, informing blue-team actions in scenarios [65]. Moreover, the **NIST Security Content Automation Protocol (*SCAP*)** [66] provides standardized specifications for expressing and sharing security content and vulnerabilities [59], [67].

Likewise, community threat-sharing platforms and schemas play a pivotal role. **AlienVault Open Threat Exchange (*OTX*)** [68], [69] is a collaborative threat intelligence platform that crowd-sources real-time Indicators of Compromise (malicious IPs, file hashes, etc.) in the form of *pulses* (threat data packages). OTX supplies up-to-date threat feeds and attack trend data that can be injected into exercises for greater realism (e.g., using current IOC lists in range networks). Similarly, **Vocabulary for Event Recording and Incident Sharing (*VERIS*)** [70] is an open framework from Verizon for standardizing how security incidents are described and shared. By using the VERIS schema (which defines incident attributes like actors, actions, assets, and impacts), scenario creators can design data-driven incidents that mirror real breach patterns, supporting pattern-based scenario design and analysis. Finally, the **NICE Cybersecurity Workforce Framework (*NIST SP 800-181*)** [71] provides a structured model for defining cybersecurity work roles and required skills. Incorporating NICE in scenario planning ensures that training objectives align with well-defined roles and competencies, e.g., mapping scenario tasks to specific SOC analyst or incident responder roles, thereby grounding exercises in workforce development goals. Each of these resources adds value to cyber range training exercise scenario generation by contributing structured knowledge. *CWE* and *CVE* bring vulnerability context, *ATT&CK* and *D3FEND* cover attacker tactics and defensive controls, *CAR* contributes detection analytics, *OTX/MISP* inject live threat data, *VERIS* provides incident patterns, and *NICE* ties scenarios to workforce roles.

E. Multi-Source CTI Integration in Scenario Generation

One emerging approach to scenario generation leverages *multi-source CTI pipelines* and knowledge-based models to synthesize rich attack-defense scenarios. In practice, multiple threat intelligence knowledge bases can be *chained* together to provide a holistic view of attacks and countermeasures. For example, an adversary technique from *MITRE ATT&CK* [17] can be mapped to a corresponding attack pattern in MITRE's **Common Attack Pattern Enumerations and Classifications (*CAPEC*) [72]–[74]**, which in turn links to known software weaknesses (*CWE* [63]) and specific vulnerabilities (*CVE* [61]), while defensive techniques from *MITRE D3FEND* [64] are associated to each stage. By linking these CTI sources, scenario designers can derive multi-step attack narratives grounded in real-world tactics, exploitable weaknesses, and mitigation measures. These linked knowledge bases are often represented through **attack graphs** or **attack trees**, formal models that capture the logical steps an adversary

could take and where defenses can break the chain. Emerging works [75] have also explored leveraging large language models and knowledge graphs to extract actionable threat intelligence from unstructured data for scenario generation.

Tools and frameworks have begun to exemplify this integrated modeling. For instance, MITRE's **Critical Infrastructure Cyberspace Analysis Tool (*CICAT*)** [76], [77] automatically generates multi-step attack scenarios by combining ATT&CK tactics with an infrastructure model of the target environment. The output is an attack graph enumerating possible attack paths, with each step annotated by the ATT&CK technique used, the relevant CVEs for exploited vulnerabilities, and potential countermeasures (ATT&CK mitigations or *NIST 800-53* [78] controls) at that step. Academic frameworks like **Generating Attack Scenarios for Cybersecurity Exercises on Industrial Control Systems (*GENICS*)** [79], focused on **Industrial Control Systems (*ICS*)** training, take a similar knowledge-driven approach where GENICS builds an augmented attack tree that maps ICS-specific MITRE ATT&CK tactics to critical assets, incorporates the corresponding vulnerabilities (CWEs/CVEs) on those assets, and even quantifies each step's risk (e.g., via *CVSS* [80] or *DREAD* [81] scoring) to produce realistic, data-informed exercise scenarios. In both cases, multiple CTI sources are fused into a unified model (graph or tree) of an attack, ensuring that each scenario is not a linear, hand-crafted script but a **structured attack–defense model** grounded in threat intelligence. Similarly, research prototypes like *CRUcialG* [82] automatically reconstruct attack scenario graphs from free-text threat reports, and a semi-automated framework [83] has been proposed to turn heterogeneous CTI reports into executable adversarial workflows. Additionally, techniques for linking fragmented CTI information to attribute **Advanced Persistent Threat (*APT*)** campaigns are being developed to streamline threat scenario creation [84].

This modeling approach offers notable *pedagogical and operational advantages*. Pedagogically, scenarios derived from CTI-integrated graphs or trees ensure that training exercises stay aligned with real, current threat behaviors and known vulnerabilities, thereby enhancing realism and relevance for the participants. Because each node in the attack graph corresponds to a well-documented technique or weakness, trainees can trace the *kill chain* of events, learning how a specific CWE/CVE is exploited by a particular ATT&CK technique and what defensive controls (e.g. D3FEND techniques) could have interrupted it. This traceability reinforces learning objectives by linking abstract threat concepts to concrete examples. Moreover, the **attack tree structure** explicitly visualizes multiple paths and branches an adversary might take, which encourages trainees to practice decision-making for different contingencies in the scenario. Operationally, integrating CTI into scenario models yields a form of automated threat modeling that can be tailored to an organization's environment. By inputting an enterprise's asset inventory and linking to CTI repositories, security teams can auto-generate attack graphs highlighting the most plausible paths an attacker might follow in that specific environment, a technique already demonstrated by *CICAT* for critical infrastructure systems [77]. These knowledge-based scenarios serve as living documents of organizational risk where defenders can not only rehearse incident response on them, but also use them to identify *hot spots* (high-risk nodes or edges in the graph) and prioritize defensive investments accordingly. Another advantage is **maintainability**, since the scenario content is derived from up-to-date intelligence feeds and standardized knowledge bases, it can be **updated systematically** as new threats emerge. For example, if a new CVE or ATT&CK technique is reported, it can be plugged into the CTI pipeline to regenerate or adapt scenarios, keeping training exercises in sync with the evolving threat landscape. To facilitate such intelligence-led updates, comprehensive datasets of APT campaigns have been published, e.g., an open repository of 86 APTs and 350 attack campaigns, and a public database of APT incidents by threat researchers [85]–[87]). Hence, multi-source CTI integration models, chaining frameworks like *ATT&CK to CAPEC to CWE/CVE* and incorporating knowledge-driven graphs/trees, enable **cyber threat intelligence-led scenario generation**. Such scenarios are not only more realistic and comprehensive (covering both attacker and defender viewpoints) but also easier to adapt and analyze, thereby greatly enhancing their educational value and operational usefulness in cyber range training environments.

F. Cyber Range Training Exercise Scenario Execution Platforms

Cyber range training exercise (CyRaTrEx) scenario execution platforms (or **execution platforms**) are the underlying infrastructure and software systems that instantiate and run cybersecurity training or evaluation scenarios. Such platforms manage the provisioning of virtual networks and hosts, deploy system images and attack tools, and generate adversarial or benign activities according to scenario scripts or live operator commands. In effect, an execution platform provides a controlled, reproducible environment in which cyber scenarios unfold and participants are assessed [88].

For taxonomy purposes, execution platforms can be grouped into **four broad categories**, reflecting their origin, target users, and design priorities, that we describe below.

1) **Academic/Research Testbeds:** These ranges are developed in universities or research laboratories for experimentation and education. They emphasize extensibility, formal scenario modeling, and integration of advanced training features (e.g., adaptive difficulty, analytics). Representative examples include ***FORESIGHT*** [89] and ***THREAT-ARREST*** [48] frameworks and the *University of Genoa's* ***CRACK*** [16], [90] platform [2], [91]. Such testbeds often prototype new ideas but may lack the polished user interfaces and dedicated support of commercial products.
2) **Open-Source Community Platforms:** These frameworks are freely available and maintained by academic and practitioner communities. They provide cyber range capabilities without licensing costs, enabling collaborative development. Open-source ranges typically use **Infrastructure-as-Code (*IaC*)** scenario definitions

(YAML/JSON, ***Terraform*** [92]) and align exercises with frameworks like MITRE ATT&CK. Representative platforms include the ***KYPO*** **Cyber Range** ***(Masaryk University)*** [50], [51] and ***MITRE's Caldera framework*** [93], [94]. They often require more technical expertise to deploy but benefit from community contributions and extensibility [27], [30].

3) **Commercial/Industrial Solutions:** These are vendor-developed platforms targeting enterprise or government customers. Commercial ranges deliver turnkey solutions with extensive content libraries, graphical scenario editors, and dedicated support. They prioritize usability, scalability, and integration with enterprise IT. Notable examples include ***Cyberbit*** [24], ***SimSpace*** [22], [95], ***Cisco*** [96], and ***Circadence's Project Ares*** [97]. These systems provide drag-and-drop scenario builders, real-time scoring dashboards, and built-in content for many threat scenarios. They are fully supported but often expensive, limiting customization [88], [98]. Additionally, domain-specific ranges have also been prototyped for industrial control systems, e.g., ***ICSrange*** [99] for **Industrial Control System/Supervisory Control and Data Acquisition (*ICS/SCADA*)** to monitor and control physical processes, ***ICSSIM*** [100], an open-source ICS simulator, and ***Estonian Cyber range*** [101] for emerging domains like *Space* domain.
4) **Government/Defense Systems:** These cyber ranges are funded and operated by national defense or intelligence agencies for critical training. They support large-scale, high-fidelity exercises (**often multi-national or classified**) and simulate critical infrastructure (including **Operational Technology (*OT*)**/ICS systems). Examples include ***Estonia's CR14 range*** [102] (hosting ***NATO's Locked Shields*** [9])), ***Sweden's CRATE*** [91], [103] (with extensive ICS integration), ***Austria's AIT Cyber Range*** [104], [105], and the ***U.S. Department of Defense's National Cyber Range (NCR)*** [53]. These platforms emphasize security and interoperability but are not open to the public and require specialized expertise to manage.

There are several dimensions with which CyRaTrEx scenario execution platforms can be described. Below, we detail **6 technical dimensions** to describe these execution platforms.

1) **Orchestration Layer:** This layer manages automated deployment of the range environment and scenarios. It includes tools for configuration management and infrastructure provisioning. Examples include ***Ansible*** [44] or ***Puppet*** [106] for host configuration, ***Terraform*** [92] or ***OpenStack Heat*** [43] for resource provisioning, and ***Kubernetes*** [107] for container orchestration [108]. A robust orchestration layer enables rapid, repeatable instantiation and teardown of exercise environments with minimal manual effort.
2) **Virtualization support:** This refers to the virtualization technology used for hosts and networks. Platforms may employ full-system virtual machines (hypervisors like **Kernel-based Virtual Machine *(KVM)*** [109] or ***VMware***) [110] or lightweight containers (e.g., ***Docker*** [111], ***Podman*** [112]), or a combination of both. **Virtual Machine (*VM*)**-based ranges emulate complete operating environments with high fidelity, while containerized services can be deployed rapidly at large scale [99]. Many modern ranges use a hybrid approach, running core network nodes in VMs and auxiliary services in containers to balance realism and performance.
3) **Scenario Injection Capabilities:** This dimension describes how adversarial or benign actions are enacted within the range. Injection methods include manual (i.e., live Red/Blue teams performing actions [113]), scripted (i.e., predefined attack/playbook scripts [114]), and API-driven (i.e., external orchestration APIs triggering events [115]). Scripted scenarios often use scheduled timelines (e.g., YAML/JSON event files) to launch attacks and events. For instance, automated bots or traffic generators may run in the background to generate realistic background traffic or orchestrate multi-stage attacks.
4) **Monitoring and Telemetry:** Execution platforms collect detailed logs of system and participant activity for scoring and analysis. Typical telemetry channels include network packet captures (via *taps* or *virtual switches*), host system logs (OS and application logs), and security agent probes such as **System Monitor (*Sysmon*)** [116] or ***Zeek*** [117] sensors where *Sysmon* focuses on what is happening on a machine, and *Zeek* focuses on what is happening on the network [5]. These data are often aggregated into dashboards or **Security Information and Event Management (*SIEM*)** to provide real-time feedback to instructors and participants. High-quality telemetry enables precise evaluation of participant actions and aids post-exercise debriefing.
5) **Scalability and Multi-User Support:** Cyber ranges differ in their ability to simulate large infrastructures and support multiple users or teams concurrently. High-end platforms can instantiate hundreds or thousands of virtual nodes, leveraging cloud elasticity, to emulate large enterprise or national-scale networks [118]. They often support multi-tenancy or parallel exercises, allowing multiple Red/Blue teams or classroom cohorts to operate in isolated subnets. In contrast, smaller academic ranges may only support a single exercise or a limited number of virtual machines at once.
6) **Support for SDL/CTI Integration:** Many modern ranges support formal SDLs and incorporate CTI feeds. SDLs (often based on *YAML*, *JSON*, or *TOSCA* schemas) allow scenarios to be defined at a high level and shared across platforms. Here, CTI integration means using threat libraries (*MITRE ATT&CK*, *CAPEC* [72]) or intelligence feeds (*MISP* [36], *STIX/TAXII* [57], [58]) to inform or tag exercise content. For example, exercises might automatically incorporate the latest threat techniques from a CTI feed, enhancing realism and enabling interoperability [18].

Representative CyRaTrEx scenario execution platforms illustrate these categories and dimensions as well. For instance, *KYPO* Cyber Range [50], [51] uses *Ansible* and *Terraform*

on an *OpenStack* cloud to automate full network deployment and telemetry, whereas MITRE's *Caldera* [93], [94] employs an agent-based framework to orchestrate *ATT&CK*-aligned attack sequences on target hosts. Commercial ranges like *SimSpace* [22], [95] and *Project Ares* [97] provide browser-based editors, large-scale network templates, and real-time scoring dashboards. Each platform exemplifies a particular set of trade-offs suited to its intended use case [27], [30].

Primarily, our taxonomy comprises of three tightly linked layers (i) ***CyraTrEx* scenario description languages (SDLs)**, (ii) **CTI-driven scenario generation pipelines**, and (iii) ***CyraTrEx* scenario execution platforms**. Across the surveyed works, implementations vary widely and lack common standards, which limits portability and reuse. To make these differences concrete, we undertake a structured comparative evaluation across these layers in the following section, i.e., Section V.

V. Comparative Evaluation

As we detail in the previous section, cyber ranges integrate multiple components, primarily SDLs, CTI-based scenario generation pipelines, and scenario execution platforms. Hence, it is crucial to systematically evaluate them in order to understand their relative strengths and limitations. This will also help in selecting appropriate components, revealing capability gaps, and guiding future research in this space. Therefore, in this section, we compare representative solutions at each layer in a structured manner. The key comparison dimensions are identified for each category as follows.

1) **CyRaTrEx Scenario Description Languages (SDLs):** Syntax type (e.g., textual vs. graphical SDLs), reusability of scenario modules, extensibility to new attack models, availability of tool support (editors, validators, exporters), and support for formal verification or static analysis.
2) **CTI-driven Scenario Generation Pipelines:** Cyber threat intelligence sources used (open feeds, structured databases like *MITRE ATT&CK* or *CAPEC*), degree of automation (manual enrichment vs. end-to-end automation), alignment with known cyber threat frameworks (e.g., mapping to *ATT&CK* tactics/techniques or *CAPEC* patterns), and scenario realism (fidelity to real-world attack behavior).
3) **CyRaTrEx Scenario Execution Platforms:** Virtualization backend (e.g., full virtual machines, containers, or hybrid solutions), orchestration tools (e.g., *Terraform*, *Kubernetes*, custom engines), native support for SDL and CTI integration, telemetry and monitoring capabilities (e.g., logging, dashboards), and overall system scalability (e.g., number of hosts, performance under load).

Our comparative methodology is multi-faceted. We compile feature matrices to capture each solution's attributes, then map these features to standard frameworks and training objectives (e.g., *ATT&CK* tactics or *CAPEC* patterns). Each criterion is applied consistently across solutions to ensure objectivity. All data comes from documented sources, including product documentation, open-source repositories, and peer-reviewed publications, in order to maintain transparency, reproducibility, and verifiability.

In the following **sub-sections V-A–V-C**, we present a structured **comparative** evaluation of the surveyed approaches, covering scenario description languages, CTI-driven generation pipelines, and scenario execution platforms. Additionally, **sub-sections V-D–V-G** extend this analysis to **complementary** aspects, including open-source adversary simulation tools, commercial and government cyber range platforms, cross-layer patterns and interdependencies, and scenario instrumentation and observability.

A. Scenario Description Languages (SDLs)

Scenario Description Languages (SDLs) provide a high-level, often declarative, means to specify CyRaTrEx scenarios. Representative SDLs include ***CRACK*** (a *TOSCA*-based language) [16], ***CAMEL*** (a model-driven cloud application SDL) [119], the **SSFnet Domain Modeling Language (*DML*)** [13], and ad hoc ***YAML***-based schemas used in platforms like ***EDURange*** [121], [122], ***Nautilus*** [123] or ***KYPO*** [124]. These languages differ in syntax, extensibility, and tool support. For example, CRACK SDL uses *YAML* [38]/*TOSCA* [45] syntax to declare virtual machines, networks, vulnerabilities and policies, enabling declarative specification of scenario elements [16]. CRACK also leverages a formal Datalog encoding to automatically verify scenario correctness against training goals [40]. In contrast, CAMEL is a textual **Eclipse Modeling Framework (*EMF*)** [120])-based SDL for modeling cloud-deployed applications (not specifically designed for attack scenarios) [119] that is highly extensible via integration of multiple SDLs, however, lacks built-in formal analysis. Legacy Domain Modeling Language (*DML*) in *SSFnet* uses an *XML* [39]-like syntax for network scenarios [13], with limited extensibility or reuse beyond its fixed schema. Simpler YAML-driven approaches (e.g. *EDURange*, *Nautilus*, *KYPO*) use structured YAML files for each scenario component and are easy to read and reuse across exercises, however they rely only on generic YAML tooling with no scenario-specific compiler or verifier [124].

Table II highlights the key trade-offs of a few representative *CyRaTrEx* scenario description languages. Declarative SDLs like CRACK enable rich expressiveness and even formal scenario checking [16], [40], but are relatively complex and verbose. CAMEL's model-driven approach offers extensibility via Eclipse/EMF but is not tailored for attack scenarios or formal analysis [119]. By contrast, YAML-based schemas are simple and human-readable, facilitating reuse of scenario fragments and portability, but they omit formal structure and rely only on general YAML tooling [124]. In practice, many modern ranges favor declarative formats (YAML/JSON, TOSCA) for ease of integration, accepting the trade-off that higher expressiveness (and tool-supported verification) comes at the cost of steeper learning and potential verbosity.

B. CTI-Driven Scenario Generation Pipelines

Modern cyber range platforms often include pipelines that transform cyber threat intelligence (*CTI*) feeds into executable training scenarios. Representative approaches include model-driven frameworks like ***CRACK*** [16], purpose-built tools such

TABLE II: Comparison of a few representative CyRaTrEx scenario description languages

CyRaTrEx Scenario Description Language (SDL)	**Syntax** (notation)	**Extensibility** (new elements)	**Reusability** (of scenarios)	**Tools** (editors, etc.)	**Formal verification**
CRACK SDL [16]	YAML [38] (TOSCA [45])	high (type derivation)	moderate	TOSCA/CRACK toolchain, YAML editors	Yes (Datalog-based)
CAMEL [119]	Textual (Eclipse Modeling Framework [120])	high (modular SDLs)	moderate	Eclipse/EMF editors	No
SSFnet Domain Modeling Language [13]	XML [39]	low (fixed schema)	limited	Legacy SSFnet tools	No
YAML [38] schemas	YAML (custom)	moderate (custom schema)	high (templates)	Any YAML editor	No

as ***SecGen*** [49], and AI-enhanced systems like ***AiCEF*** [18] or LLM-based generators like ***ARCeR*** [125]. For example, the CRACK framework provides an end-to-end scenario development pipeline where it uses a **TOSCA-based Scenario Definition Language *(CRACK-SDL)*** to describe network infrastructures, applies formal validation (e.g. Datalog queries) for correctness, and then auto-deploys the scenario to an **Infrastructure as a Service (*IaaS*)** platform [16]. While CRACK automates design and deployment, it relies on fixed component libraries (e.g. CAPEC/CVE entries) rather than dynamic feeds. In contrast, ***Filigran's OpenBAS platform*** [19] closely integrates with the ***OpenCTI threat repository*** [126], where it can pull live intelligence (MITRE ATT&CK TTPs, MISP data, malware reports, etc.) and automatically synthesize corresponding attack injects into a scenario. Similarly, ***AiCEF*** [18] uses NLP and clustering on unstructured cyber incident reports to tag scenario elements (via a *Cyber Exercise Scenario Ontology*) and generate structured exercises with minimal human input. For instance, the ***CRUcialG*** [82] system automatically constructs attack scenario graphs from CTI reports and verifies their coherence. Finally, emerging pipelines like ***ARCeR*** [125] employ large-language models with retrieval augmentation to accept high-level natural language (*English*) scenario descriptions and produce ready-to-deploy range configurations. Large-language models combined with knowledge graphs can similarly assist CTI extraction [75]. These systems vary widely in their use of CTI, as *CRACK* and *SecGen* use only static *CVE* and *CAPEC* libraries, *OpenBAS* pulls real-time *ATT&CK* and *MISP* feeds, and *AiCEF* and *ARCeR* leverage textual reports or prompt-based knowledge.

Table III summarizes the aforementioned scenario generation pipelines. We observed that ***greater automation often comes at a cost to fidelity or transparency***. For instance, SecGen and CRACK can generate complex networks with little manual work, but SecGen's randomness yields low realism, and CRACK's formal models require manual threat injection. OpenBAS and AiCEF embed real threat data to boost realism, but still rely on template injects or NLP-derived narratives, so human review is typically needed to ensure consistency. In most cases, mapping raw CTI (e.g., **Indicators of Compromise (*IOC*)** lists, unstructured reports, etc.) to low-level scenario actions remains limited, as pipelines focus on known TTP libraries (e.g., *ATT&CK IDs*) or fixed exploit sets, which may not capture the full context of an attack. Moreover, the tool support varies, e.g., CRACK and OpenBAS provide end-to-end exporters and configuration tooling, whereas AI-driven frameworks (AiCEF, ARCeR) are still mostly research prototypes with ad hoc scripts. Hence, existing systems illustrate ***trade-offs between manual tailoring and automation***. Automated CTI-to-scenario methods can rapidly adapt scenarios to emerging threats, but they face gaps in semantic interoperability, fidelity of attacker behavior, and ease of use. Addressing these gaps, such as standardizing data mappings or augmenting automated outputs with expert checks, is an ongoing challenge in CTI-driven design.

C. CyRaTrEx Scenario Execution Platforms

Several CyRaTrEx (i.e., Cyber Range Training Exercise) Scenario Execution Platforms have been developed in academia, industry, and government. For example, ***KYPO*** [124] (by *Masaryk University, Czech Republic*) is an open-source, cloud-oriented scenario execution platform built with microservices and Infrastructure-as-Code (*IaC*). It uses *OpenStack* [43] for Virtual Machines (*VMs*) and network virtualization and *Docker* [111] containers for services. ***DETER-Lab*** [28] (*The United States' National Science Foundation's testbed*) is a hybrid academic/government platform based on the *Emulab* [128] framework. It supports multi-resolution virtualization (VMs, lightweight containers) and provides a **Graphical User Interface (*GUI*)** or **Command Line Interface (*CLI*)** for orchestrating large-scale experiments. Among commercial ranges, ***Cyberbit*** [24] is a **Software-as-a-Service (*SaaS*)** that spins up realistic networks (on-premises, cloud or hybrid) for live exercises. It emphasizes hyper-realistic scenarios, integrates standard tools such as **Security Information and Event Management (*SIEM*)**, firewalls, etc., and maps to frameworks like *ATT&CK* and *NICE* [71]. Another platform ***RangeForce*** [25] is a cloud-based training range offering on-demand exercises with real security tools that highlights team performance metrics and alignment to *MITRE* and *NIST* however, is proprietary. **SCYTHE Breach and Attack Simulation+ (*BAS+*)** [129] is a commercial

TABLE III: Comparison of a few representative CTI-driven scenario generation pipelines.

Pipeline	CTI Sources	Automation	Fidelity	Standards Mapping	Tool Support
CRACK [16]	Static libs (CAPEC [72], CVE [61])	High (automated design, verification, deploy)	Medium (formal-structure, real networks but scripted attacks)	Uses CRACK SDL, can reference CAPEC/CWE [63]	CRACK toolchain (SDL, IaaS, Datalog verifier)
SecGen [49]	CVE/exploit databases	High (random scenario synthesis)	Low (randomized VMs, little attacker context)	Limited (No ATT&CK/CAPEC integration)	Standalone generator, scripts (VM builder)
OpenBAS [19]	ATT&CK, MISP, malware lists via OpenCTI [126]	High (one-click CTI-to-scenario)	Medium-High (TTP-driven inject sequences)	Maps threat reports to ATT&CK TTP injects, exports to JSON/YAML	OpenBAS UI/API, OpenCTI connector, inject exporter
AiCEF [18]	Unstructured sources (news, reports, etc.)	High (NLP / ML pipeline)	Medium (narrative-driven, expert review needed)	Uses custom ontology, can tag to ATT&CK or CAPEC patterns	Python NLP pipeline, ontology database, scenario builder
ARCeR [125]	LLM text prompting	High (LLM+Retrieval Augmented Generation (RAG [127]) auto configuration)	High (leverages documentation for realistic configs)	Not focused on TTP mapping, produces infrastructure specs.	LLM-agent support, vector DB, config checker

adversary-emulation platform using *VMware* (VM-based lab environments) to deliver isolated training scenarios for **red team (*attackers*) and blue team (*defenders*)**. At the government level, national ranges such as ***US National Cyber Range*** [53], are typically closed systems deployed on private infrastructure for high-fidelity emulation and large-scale exercises.

Table IV draws a comparison of these platforms based on the identified properties during our review. In the following text, we elaborate these properties in detail.

1) **Deployment and virtualization.** Academic and government ranges (*KYPO [124], DETERLab [28], US National Cyber Range [53]*) often run on private or academic cloud platforms and support both VM-based and lightweight container emulation. Commercial ranges (*Cyberbit [24], RangeForce [25], SCYTHE BAS+ [129], DIATEAM [131]*) generally use cloud-based or hosted VM infrastructures for scalability and ease of access. For example, *DIATEAM* offers hybrid cyber ranges that can be deployed on-premises, mobile or via service. Moreover, containerization is growing (*KYPO* and *Airbus* [56] use *Docker*, *Terraform/Ansible*, etc.), however, many platforms still rely on traditional VMs or even physical hardware for full-system fidelity.
2) **Automation and scenario injection.** Modern ranges increasingly automate setup via scripts or *CI/CD* pipelines. *KYPO* and *AIT* [105] use Infrastructure-as-Code (*IaC*) tooling (e.g., *Ansible/Terraform*) to provision networks and hosts [51], [104]. Commercial solutions such as *Immersive Labs* [97] and *Keysight* [26] often provide libraries of pre-built scenarios and web interfaces for deployment. Although, scenario injection varies as many systems still require manual or scripted insertion of attack events (e.g., SCYTHE BAS+ with workbooks, custom *DIATEAM* scenarios), though some offer APIs or integrated threat libraries. For instance, *Cisco Talos Range* [96] delivers expert-designed attack exercises driven by real threat intelligence. Notably, research/government ranges such as *US National Cyber Range* and *CITEF* [134] aim for fully scriptable and repeatable exercises using timeline-based scenarios.
3) **Instrumentation fidelity.** Open-source and research ranges prioritize fidelity, e.g., *DETERLab* can capture full packet traces and telemetry (supporting 100k-node Distributed Denial of Service (*DDoS*) experiments) [26]. *KYPO* [124] and *AIT* [105] expose system logs and allow monitoring of guest VMs [56]. On the other hand, commercial ranges emphasize actionable metrics and higher-level feedback relevant to learners, e.g., detection rates, response times, skill scores, etc. For instance, *Immersive Labs* provides data-driven benchmarks and team performance insights [97]. In general, higher-fidelity instrumentation (e.g., complete Packet Capture (*PCAP*), hardware timing, etc.) comes at the cost of complexity and is more common in research/government setups, whereas commercial products report aggregated results and leaderboard scores.
4) **Accessibility.** Platforms differ in access, e.g., *KYPO* and *SANS NetWars* [130] are open to broad audiences, where *KYPO* is MIT-licensed [50], *SANS* provides public Cyber Ranges under commercial license [131]), and *AIT*'s range is used for national-scale training [105]. *DETERLab* and *DIATEAM* require contracted use, while commercial ranges such as *Cyberbit*, *RangeForce* [25], *CybExer* [132], *Nortal Coliseum* [133], and *Keysight* [26]), generally require licenses or subscriptions. Government ranges such as *US National Cyber Range* and *CITEF*, are closed to authorized users only. This reflects a trade-off between openness and capability where open-source platforms foster collaboration and extensibility, while proprietary/government systems may have richer features with limited sharing.

TABLE IV: Comparison of a few representative Scenario Execution Platforms

Platform	Deployment Environment	Virtualization Technology	Automation	Instrumentation	Scenario Injection	Access
KYPO CRP [50]	Cloud (OpenStack [43])	VMs + Containers	Scripted (Ansible), Web UI	System logs, metrics	GUI / Scripts	Open-source (MIT)
DETERLab [28]	On-prem (Emulab [128])	VMs + Containers	CLI orchestration	Full PCAP, logs	Experiment scripts	Open (academic)
Cyberbit Range [24]	SaaS (AWS/cloud)	VM-based networks	Pre-built scenarios	Team metrics (KPIs), event logs	Web UI	Licensed (commercial)
SCYTHE BAS+ [129]	Cloud (VMware Lab)	VMs (isolated labs)	Lab reset / management	Training console logs	Manual (workbook-driven)	Licensed (commercial)
RangeForce Cloud [25]	SaaS (cloud)	VMs/Containers (cloud)	UI-driven deployment	Team performance data	Web UI/API	Licensed (commercial)
US National Cyber Range [53]	Private (Govt)	Hybrid (VM + Hardware)	Custom orchestrator	High-fidelity (PCAP/logs)	Automated scripts	Restricted (government)
Airbus CyberRange [56]	SaaS/Cloud (Airbus)	VMs + Containers	Web UI/ Orchestrator	System logs, metrics	GUI-driven	Licensed (commercial)
Cisco Talos Range [96]	Cloud (Cisco)	VMs	Pre-built scenarios	Team metrics	Expert-driven scenarios	Licensed (commercial)
Immersive Labs [97]	SaaS (cloud)	VMs + Containers	UI-driven deployment	Team performance data	Web-based labs	Licensed (commercial)
Keysight Cyber Range [26]	On-prem (Keysight)	VMs + network sim.	Scenario builder	System logs, traffic metrics	Real-world traffic injection	Licensed (commercial)
SANS NetWars [130]	Cloud (SaaS)	VMs	Pre-defined labs	Skill scorecard	Gamified challenges	Licensed (commercial)
DIATEAM Cyber Range [131]	On-prem / SaaS	VMs + ICS/SCADA	Scenario designer	Logs, metrics	Manual (workbook)	Licensed (commercial)
CybExer [132]	SaaS (cloud) / On-prem	VMs (digital twin)	Web UI orchestrator	Team performance data	Pre-built scenarios	Licensed (commercial)
Nortal Coliseum [133]	SaaS/ On-prem	VMs + digital twin	Orchestration	High-fidelity logs	Automated scripts	Licensed (commercial)
CITEF [134]	Private (Govt)	VMs (digital twin)	Timeline-based scripts	Metrics, logs	Automated scenarios	Restricted (government)
AIT Cyber Range [105]	On-prem (OpenStack)	VMs + containers	IaC (Terraform + Ansible)	System metrics, logs	Scripted scenarios	Restricted (research)

D. Open-Source Adversary Simulation Tools

In **addition** to the open-source platforms mentioned in **Table IV**, several **other** open-source adversary simulation frameworks can augment cyber range exercises by emulating realistic attacker behaviors and attack sequences. These include ***Metta*** [135], ***Atomic Red Team*** [136], ***Infection Monkey*** [137], ***DumpsterFire*** [113], **Red Team Automation (*RTA*)** [138], and ***Stratus Red Team*** [139]. Each tool provides distinct capabilities for injecting adversarial activities into training scenarios, similar to the commercial *SCYTHE BAS+* [129]. Below, we elaborate on these **adversary simulation** capabilities.

1) **Adversary Behavior Emulation:** Tools like *Metta* and *Infection Monkey* automate multi-step attack campaigns. *Metta* is a preparedness framework that runs adversary actions defined in YAML (mapped to MITRE ATT&CK) to test host and network detection capabilities [135]. *Infection Monkey*, a breach-and-attack simulation platform, releases self-propagating "monkeys" that attempt exploitation and lateral movement across the network, revealing security gaps by aggressively spreading and "running amok" in target environments [137], [140]. Both enable red teams to emulate realistic post-compromise behaviors without manual intervention.
2) **Noise Generation and Event Sequencing:** *DumpsterFire* focuses on generating synthetic security events and noise in a controlled timeline. It allows users to build custom "incidents in a box", time-delayed chains of benign and malicious events, to simulate insider threats, decoy attacks, or distractions for blue teams [113]. This helps exercise participants practice distinguishing true attacks from background noise and incremental events in a Security Information and Event Management (*SIEM*) or logging environment.
3) **Cloud Environment Attack Simulation:** *Stratus Red Team* extends adversary emulation into cloud platforms. Described as "Atomic Red Team for the cloud," it provides a collection of atomic attack techniques tailored to cloud services (*AWS* [141], Azure [142], GCP [143],

Kubernetes [107]). Security teams can use Stratus Red Team to simulate cloud-specific threats (e.g., privilege escalation, cloud persistence tactics) in order to validate cloud monitoring and response processes.

4) **Atomic Test Libraries:** *Atomic Red Team* and *Red Team Automation (RTA)* offer curated sets of atomic test cases for adversary tactics. *Atomic Red Team* is a widely used library of simple, discrete attacks (mapped to ATT&CK) that can be executed one by one or tied together to test detection of individual techniques [136]. Similarly, Red Team Automation *(RTA)* is an open-source script repository that implements dozens of attacker TTPs as small "test scenarios" [138]. While neither provides a full campaign out-of-the-box or autonomous orchestration, they equip range operators with repeatable building blocks (scripts, command sequences) to inject specific tactics into an exercise on demand.

Integrating these open-source tools into a cyber range platform can enrich the exercise realism. They allow range operators to ***simulate adversary behavior at varying scales***, from noisy background traffic generation to stealthy post-exploit actions, and to cover attack vectors in both traditional IT networks and cloud environments. By leveraging frameworks like *Atomic Red Team* and *RTA* for atomic checks, and tools such as *Metta*, *Infection Monkey*, and *DumpsterFire* for broader attack narratives, instructors can create more dynamic, threat-informed scenarios that evaluate detection and response across a spectrum of conditions. This can complement the built-in capabilities of a cyber range execution platform and can ensure that exercises stay aligned with real-world tactics and emerging threats.

E. Commercial and Government Cyber Range Platforms

As illustrated in **Table IV**, several notable cyber range solutions, both from the industry and the government, enrich the cyber range ecosystem. Below, we elaborate on these solutions and what they offer.

1) **Airbus CyberRange,** an integration and simulation platform by Airbus that allows building complex virtual and physical systems to simulate realistic scenarios [56]. It provides a high-fidelity environment for advanced cyber exercises, emphasizing realistic network and system replication.
2) **Cisco Talos Range,** a cyber range service by Cisco's Talos Intelligence Group focused on incident response training driven by real threat intelligence [96]. This platform provides immersive, expert-designed exercises using up-to-date Cisco threat intel to simulate real-world attacks and defenses, enabling teams to improve detection and response skills.
3) **Immersive Labs,** an interactive training platform offering hands-on cyber range exercising through browser-based labs and team challenges [97]. It provides on-demand gamified exercises mapped to frameworks like *MITRE ATT&CK*, delivering real-world skill development and team performance benchmarking in a safe environment.
4) **Keysight Cyber Range,** a simulation-driven range solution from Keysight that reproduces real-world network traffic and attack scenarios for enterprise security training [26]. Built on Keysight's network test expertise, such as *BreakingPoint* traffic generator [144], it offers a scalable environment with customizable network models, realistic attack injection, and high-fidelity monitoring to validate defenses under realistic conditions.
5) **SANS NetWars,** a suite of gamified cyber defense exercises provided by the SANS Institute [130]. NetWars includes multi-level challenge ranges across domains such as forensics, penetration testing, and Industrial Control Systems *(ICS)*, with scoring and leaderboards. It delivers hands-on, interactive challenges so practitioners can sharpen their skills through realistic scenarios in a competitive setting.
6) **DIATEAM Cyber Range Solutions,** a set of customizable cyber range platforms from DIATEAM (*France*) developed for civil and military use cases [131]. Established in 2002, DIATEAM's solution provides high-fidelity, tailor-made environments, including ICS/SCADA scenarios, that can be deployed on-site or via mobile units, enabling large-scale, realistic exercises for government agencies and enterprises.
7) **CybExer,** a cyber range platform by CybExer Technologies (*Estonia*) available in SaaS and on-premise formats [132]. It focuses on improving organizational cyber resilience by allowing users to securely test their systems and train staff in realistic scenarios. The platform supports scalable, cloud-based exercises using *digital twin* emulation and has been used in NATO-aligned training programs.
8) **Nortal Coliseum,** an advanced, modular cyber range platform developed by Nortal's Talgen cybersecurity division [133]. Coliseum leverages a digital twin simulation platform to replicate an organization's IT/OT environment for high-pressure incident response simulations. It is designed for nation-scale and specialized scenarios, such as space sector drills, and has been deployed in international cyber training centers to enhance cyber resilience.
9) **Cybersecurity Intelligence Training and Evaluation Framework *(CITEF)*,** a cyber range originally developed by RHEA Group (now operated by Starion Group's Nexova division) for realistic training and assessment in IT and industrial networks [134]. CITEF enables users to create digital twins of their environments and run timeline-based attack scenarios. It offers extensive libraries of assets and threats, supports multi-tenant use for concurrent exercises, and provides performance evaluation tools in cyber drills.
10) **Austrian Institute of Technology *(AIT)* Cyber Range,** an Austrian research-based cyber range platform [105], offers a virtual environment for simulating critical IT and industrial control system scenarios. Built on a modular open-source architecture such as OpenStack, Terraform, and Ansible, it supports up to dozens of participants on-site. The AIT range provides hands-on exercises including ransomware and ICS-specific attacks, in a controlled,

TABLE V: Interdependencies among the 3 layers (scenario description, generation, and execution)

Layer	Key Capability	Interdependency / Impact
SDL (*Scenario Description Language*)	Formal semantics, infrastructure + narrative modeling, interoperability	Enables automation of deployment and runtime (attacks/injects). Lack of standard SDL means CTI mapping and cross-platform reuse are manual and error-prone.
CTI (*Cyber Threat Intelligence*)-Driven Generation Pipeline	Ingests threat data (*ATT&CK, CVEs, MISP*) to generate scenario content	Requires expressive SDL to encode CTI (e.g. TTP-to-scenario mapping). Without it, generation defaults to human-in-the-loop design. Also depends on the execution platform's ability to enact CTI-informed actions.
Scenario Execution Platform	Automatic deployment, attack injection, monitoring, scoring	Relies on SDL for instructions, limited SDL semantics restrict platform automation. Platform feedback and monitoring can refine generation pipeline and SDL (e.g., validating scenario feasibility). Manual platforms inflate SDL gaps (e.g., static scripts) and hamper dynamic CTI updates.

high-realism setting, and supports the national-scale cyber defense training.

F. Cross-Layer Patterns and Interdependencies

The comparative analysis reveals that the three layers, scenario description languages (SDLs), CTI-driven generation pipelines, and scenario execution platforms, are tightly interlinked. A recurring pattern is that ***limitations in one layer propagate into the others***. For example, most existing **SDLs are informal** or ad-hoc (e.g., simple YAML/JSON schemas) and often **capture only network topology** or infrastructure, not the full exercise narrative or training objectives. This lack of formal semantics in the SDL layer **impedes automation downstream**. Without a precise, expressive language (such as *TOSCA*-based models), mapping threat intelligence (e.g., MITRE ATT&CK tactics, CVEs) into concrete scenario elements **requires extensive manual translation**. Conversely, if a CyRaTrEx scenario execution platform expects attack injections, scoring rules, or timeline triggers, these **must be encoded in the SDL**, otherwise the platform must rely on human controllers. In practice, this reflects that ***weakly-specified SDLs lead to static, hand-crafted exercises*** where instructors script scenario flow by hand and execution is orchestrated manually, rather than end-to-end automated.

At the system level, we observe a ***convergence on common frameworks*** and a ***persistent end-to-end automation gap***. Many tools and studies emphasize *MITRE ATT&CK* as a unifying reference as threat actions in scenarios are increasingly tagged with *ATT&CK* tactics/techniques, and pipelines often leverage *ATT&CK* or *STIX* for contextualizing attacks. This trend toward ***shared semantics helps bridge layers*** as SDLs that can express *ATT&CK* techniques facilitates CTI mapping and platform consistency. However, we find that fully automated pipelines remain harder to achieve. Most cyber ranges today still ***inject CTI only indirectly*** where feeds from *MISP* or *CVE* databases ***require to be curated and hand-mapped*** to scenario templates. Moreover, the scenarios themselves are generally static once deployed. In other words, even though all three layers are drifting toward common standards (e.g., ATT&CK or IaC languages), no complete and automated pipeline reliably transforms raw CTI into a running exercise without human intervention. This lack of automation remains a limitation across layers.

Table V illustrates these interdependencies at a high level. As shown, the expressiveness of an SDL has a direct impact on both pipeline capabilities and platform automation. An SDL that can encode temporal logic, user roles, and vulnerabilities, allowing the scenario generation tools to assemble attack chains aligned with CTI and enabling the scenario execution engines to autonomously stage attacks. By contrast, ***a minimal SDL forces the CTI-to-scenario mapping to be manual***, and the platform to treat the scenario as a **static configuration**. Similarly, advanced generation methods (e.g., replay-based or AI-driven scenario creation) depend on platforms that support dynamic injects and monitoring, where if the platform layer cannot ingest live feeds or adapt at runtime, then innovative pipelines cannot be fully realized.

G. Scenario Instrumentation and Observability

To fully evaluate training effectiveness, modern cyber ranges embed rich scenario instrumentation encompassing logging, telemetry collection, performance metrics, and in-scenario observability tools. In practice, comprehensive logging and telemetry frameworks (e.g., network monitors and *SIEM* pipelines) continuously capture system events, network traffic, and user actions during an exercise. This real-time data stream provides runtime visibility into the scenario's state, enabling dynamic feedback and automated scoring of participant actions [31]. For instance, open-source monitoring tools such as *Zeek* [117] have been used to instrument CyRaTrEx scenarios, offering detailed logs of attacks detected, commands executed, and system responses, i.e., information that supports real-time scoring of trainees as well as thorough after-action analysis for debriefings. Such telemetry-driven observability allows instructors and range systems to pinpoint ***how and when a trainee responds to incidents***, providing objective evidence of skills applied or gaps needing remediation. Also, recent surveys [33], [55] identify ***integrated monitoring and logging as key capabilities*** of advanced cyber ranges, underlining their role in capturing quantitative performance indicators during exercises.

Equally important are the performance metrics and in-scenario observability techniques derived from this scenario instrumentation. Cyber ranges now commonly define metrics like detection time, incident resolution speed, success/failure

rates for specific tasks, and resource utilization, all drawn from instrumented scenario data. By embedding sensors and telemetry agents within the simulated environment, the range can automatically track when critical scenario events occur (e.g., a flag compromise or malware outbreak) and ***whether the trainee's actions mitigated them in a timely manner***. These metrics feed into scoring algorithms and dashboards that give ***immediate feedback to learners***, e.g., awarding points for isolating an infected host or subtracting points for missed alerts. Such data-driven scoring and feedback mechanisms carry significant pedagogical value as trainees receive objective assessments and personalized insights into their performance, while instructors can tailor debriefings based on concrete evidence of what transpired [55]. Moreover, the collected telemetry supports common trend analytics (e.g., comparing performance across sessions or identifying common failure patterns), which is invaluable for curriculum improvement and adaptive training design. Operationally, robust instrumentation also helps cyber range operators ensure scenario fidelity and smooth operation, as real-time telemetry can flag environment issues or unintended trainee behaviors, allowing prompt adjustments during an exercise. Hence, in addition to enhancing training effectiveness through better measurement and feedback, integrating logging, telemetry, and observability into CyRaTrEx scenarios also provides a foundation for continuous improvement of the training exercises [29].

VI. Open Challenges

A. Scenario Description Languages (Layer 1)

At the SDL layer, our survey revealed two pressing challenges, i.e., C_1 **Formal Semantics and Verification**, and C_2 **Behavioural Fidelity**. We expand on these challenges below.

C_1 **Formal Semantics and Verification.** Despite numerous proposals, only *2 out of* all surveyed SDLs provide a rigorous, machine-checkable grammar or formal semantics for scenario descriptions. In the *absence* of a **formal specification**, advanced automated reasoning remains infeasible, e.g., one cannot easily verify whether every defined attack step is reachable or ensure the scenario has no *dead-ends*. This gap undermines the reliability and reuse of scenarios. A formally defined SDL would enable verification of scenario properties (soundness, completeness) before deployment [14]. For instance, *Costa et al.* [40] introduce a Virtual Scenario Description Language (*VSDL*) that translates high-level scenario definitions into logical constraints, allowing a Satisfiability Modulo Theories (*SMT*) solver to check consistency and executability. Such approaches demonstrate the potential of formal methods in this domain. However, most current languages remain informal, relying on ad-hoc validation. Future work should investigate compiling SDL specifications into verification frameworks, such as **Temporal Logic of Actions Plus (*TLA*$^+$)** [145], [146] and **Alloy** [147], [148], to systematically eliminate ambiguous semantics and enable machine-assisted scenario validation.

C_2 **Behavioural Fidelity.** Most existing SDLs emphasize adversary tactics and techniques (often tagging steps with *MITRE ATT&CK* IDs) but *fall short* in encoding the **procedure-level** detail behind those techniques. In other words, an SDL might specify that an occurred *Privilege Escalation*, but not the exact commands, exploits, or artifacts involved. This abstraction **limits realism**; trainees face a generic tactic rather than a concrete instance of that tactic. Bridging this fidelity gap requires richer scenario semantics and possibly automation. One approach is to extend the **scenario ontology to capture procedures and effects** (e.g., including specific malware behaviors or tool outputs for a given technique). Notably, the ATT&CK framework itself provides only high-level examples of attacker procedures and not exhaustive implementations [17], [93], so additional knowledge bases or domain-specific extensions (such as a *malware behavior catalog*) are needed. Another complementary approach is **automated script generation**, where recent research suggests using program synthesis or AI to convert high-level technique descriptions into concrete attack scripts and events. Early prototypes (e.g., using large language models to generate attack playbooks from CTI feeds) indicate feasibility; however, this remains an open research frontier. Thus, future SDLs **should** strive for higher behavioral fidelity, either through more expressive language constructs or integration with automated threat emulation tools, so that scenarios not only enumerate tactics but also instantiate realistic attack *procedures* and observable effects for each step, thereby offering realistic training experiences.

B. CTI-Driven Scenario Generation Pipelines (Layer 2)

Integrating Cyber Threat Intelligence *(CTI)* into scenario generation promises more dynamic and realistic exercises, but it also introduces several open research challenges. In particular, we highlight two key main challenges as C_3 **Trustworthiness of CTI Data** and C_4 **Continual Scenario Evolution** that must be addressed to fully realize CTI-driven scenario generation pipelines. We expand on these challenges below.

C_3 **Trustworthiness of CTI Data.** A fundamental challenge is ensuring the reliability and relevance of external threat intelligence data used for scenario creation. Threat feeds and intelligence reports vary widely in accuracy, timeliness, and completeness, and there is currently ***no automated, standardized way to assess or guarantee the quality of incoming CTI data***. This uncertainty in data trustworthiness indicates that scenarios risk being built on inaccurate or **outdated** threat information, which can undermine the fidelity of training exercises. Research efforts have begun proposing **CTI quality metrics** (e.g., source reliability, indicator accuracy, relevance) [149], but integrating such assessments into an automated pipeline remains an open problem [150]. A related challenge is **origin and context**, understanding where the CTI originates and under what conditions it was observed. Without robust methods to vet and annotate CTI (e.g., confidence scores

or source credibility indicators), scenario generators must either involve human analysts in the loop or risk introducing misleading artifacts. Thus, developing techniques to ***automatically evaluate and filter CTI for reliability***, while preserving actionable details, is an important research direction.

C_4 **Continual Scenario Evolution.** Despite advances in cyber ranges, transforming CTI into executable scenarios remains largely manual and labor-intensive, often requiring scripting and expert intervention at multiple stages. Fully automated generation of scenarios from structured threat intelligence is still **rare**, and human operators typically must curate intelligence, map it to scenario elements, and validate the results [40]. This gap arises from difficulties in ***translating high-level threat descriptions*** into technical actions, configuring realistic attack sequences dynamically, and ensuring pedagogical coherence [15]. Bridging this automation gap requires developing frameworks leveraging AI planning or template-based generation to interpret CTI into actionable scenario steps. Advances are needed not just in threat-data parsing but also in orchestration tools capable of deploying and adjusting scenarios in near-real time with intelligence updates.

Moreover, maintaining **temporal relevance** poses a significant challenge due to the ***continuously evolving threat landscape***. Scenarios often become stale, updated manually only sporadically, creating a lag between observed adversary tactics and training content. Incorporating newly discovered vulnerabilities, Indicators of Compromise (*IoCs*), or Tactics, Techniques, and Procedures (*TTPs*) seamlessly and continuously into CTI-driven pipelines without heavy manual effort remains an open research area [14]. The issue of **semantic drift**, where evolving threat intelligence frameworks and terminologies cause mappings between CTI data and scenario elements to lose alignment, further complicates scenario maintenance. Diverse CTI sources and evolving standards, such as new ATT&CK techniques and re-classifications, worsen semantic mismatches [93]. The solutions proposed include developing common ontologies or intermediate representations and robust versioning mechanisms in scenario description languages to accommodate evolving threat data; however, mechanisms ensuring semantic consistency remain underdeveloped [151].

Additionally, *CyRaTrEx* scenarios and their definitions require **dynamic lifecycle management**, evolving alongside rapid changes in the cybersecurity landscape. Current platforms offer **limited** mechanisms for scenario updates and evolution, making it challenging to keep content current without extensive manual redesign and maintenance. Ideally, scenario definitions would integrate threat intelligence updates (e.g., new CVEs or APT tactics) automatically, reflecting changes dynamically in the execution environment. Effective lifecycle management frameworks should treat scenarios as living entities with version control, modular updates, feedback loops, and partial modification capabilities **without rebuilding** entire environments. Open-source and academic initiatives increasingly emphasize extensibility, advocating for cyber ranges as ecosystems that evolve with cybersecurity developments [151]. Future research directions can be foreseen to include exploring DevOps-like scenario management practices, continuous integration of CTI feeds, automated testing for scenario integrity, and periodic platform updates. Moreover, ensuring **reproducibility** and **auditability** despite changes is also critical [150]. Coordinated efforts across multiple layers (i.e., CyRaTrEx SDLs, CTI-guided scenario generation, and execution platforms) are necessary to support scenario evolution sustainably. Addressing lifecycle management comprehensively can significantly **reduce manual overhead** and ensure scenarios remain continuously **relevant**, realistic, and pedagogically valuable.

C. Execution/Cyber Range Platforms (Layer 3)

While earlier sections examined the state of Scenario Description Languages (SDLs) and CTI-driven scenario generation pipelines, several fundamental issues remain unsolved at the scenario execution platform layer. We highlight two pressing research challenges, C_5 **API-Driven Scenario Ingestion** and C_{5-6} **Scalable Telemetry and Observability**, that must be addressed in future platform designs. We expand on these challenges below.

C_5 **API-Driven Scenario Ingestion** An ongoing limitation is the absence of standardized interfaces and formats for injecting scenario definitions and threat intelligence into diverse cyber ranges. Each platform today tends to be a self-contained ecosystem with proprietary APIs and ad-hoc scenario description formats. This fragmentation means that a scenario specified in one system (e.g., a YAML/JSON topology for *KYPO* [124] or a *TOSCA*-based model in *CRACK* [16]) cannot be directly imported into another, severely hampering interoperability and content reuse [33]. *Tarman et al.* [152] showed that even reproducing the ***same experiment on different range platforms can yield contrasting results***, underscoring the need for standards. While **CITEF initiative's 2024 report** [153] documents unified requirements for cyber range services to improve interoperability, and frameworks like **CyRIS** [124] proposed a common system to instantiate cyber ranges from high-level specifications, but most open-source ranges still lack standardized APIs or data exchange formats. Similarly, they **do not expose common data exchange mechanisms** or normalized telemetry formats for scenarios. As a result, integrating external tools or CTI feeds (e.g., ingesting real threat intelligence from *MISP* or *ATT&CK*) into the exercise environment **requires significant custom engineering** and format translation on a per-platform basis. Prior attempts like **ADLES** [154] and federated efforts such as **ECHO project's unified SDL** [48] underscore the community's recognition of this problem; however, **no actual standard** has yet been widely **adopted**. Establishing standard scenario injection interfaces (potentially building on existing standards or APIs) remains an open research challenge vital for en-

abling true cross-platform scenario portability and automated CTI-driven scenario updates.

C_6 **Scalable Telemetry and Observability.** A significant challenge in cyber range environments involves comprehensive instrumentation for monitoring, analysis, and maintaining a balance between simulation fidelity and scalability. Current platforms typically offer **limited built-in** capabilities for capturing detailed exercise telemetry, performance metrics, and user activities, forcing instructors to rely heavily on **external tools or manual observations** to evaluate participant responses and scenario effectiveness [5]. There exists a noticeable shortage in monitoring and analytics, as only a few ranges automatically collect comprehensive data on attacker/defender actions or provide **real-time feedback**, impacting both training effectiveness and system optimization. Moreover, telemetry data collected is often proprietary, restricting cross-platform comparisons and integration with external tools [99].

Cyber range platforms also face inherent **trade-offs** between simulation fidelity and scalability [30]. **High-fidelity** simulations employing detailed virtual machines, precise network emulation, and realistic user behavior enhance realism but incur **substantial performance overhead**, limiting scalability. For instance, VM-based infrastructures commonly used in academic and open-range contexts can emulate complex environments but suffer from prolonged startup times, high resource demands, and practical limitations on concurrent nodes or trainees. Conversely, platforms **prioritizing scalability** through lightweight containers or abstracted simulations reduce resource usage but **sacrifice realism**, omitting critical low-level system details and complex behaviors [27].

Balancing telemetry observability, fidelity, and scalability calls for novel hybrid architectures combining virtualization and lightweight simulation, smarter resource management strategies such as elastic cloud scaling or HPC backend integration, and advanced abstraction techniques to preserve essential realism at scale. Addressing these linked challenges demands **standardized** telemetry pipelines and instrumentation frameworks that embed seamlessly into cyber range environments with minimal performance overhead. Future platforms might integrate comprehensive logging, real-time dashboards, and even gamification or biometric monitoring by design. For instance, **SCORPION cyber range prototype** [155] includes adaptive gamified exercises and collects telemetry such as ***participants' heart rates for learning analytics***. Developing such scalable observability solutions remains an active area for innovation, essential for robust scenario evaluation, adaptive scenario management, and sustainable cyber range operations.

D. Cross-Layer Challenges

A number of open research challenges cut across the scenario description layer, the CTI-driven scenario generation pipeline, and the scenario execution platform. Unlike isolated issues confined to a single component, these *cross-layer challenges* demand holistic solutions spanning the entire cyber range ecosystem. We detail below a cross-cutting challenge C_7 **Reproducibility and Benchmarking**, which reflects gaps that hamper end-to-end effectiveness and interoperability of cyber threat-informed cyber ranges.

C_7 **Reproducibility and Benchmarking.** A challenge in CyRaTrEx scenarios is ensuring ***standardization, reproducibility, and portability*** across diverse platforms. Currently, platforms often rely on proprietary SDLs or formats, creating fragmentation and significantly limiting interoperability [33]. The absence of universally accepted scenario schemas reflects that ***scenarios designed for one platform cannot be easily transferred or executed elsewhere without extensive manual effort***. This lack of standardization hinders content sharing, reuse, and cyber range federation, forcing organizations to **recreate** scenarios repeatedly for different systems. Initiatives such as **European ECHO project** [156] propose unified scenario specification approaches to bridge this gap, but widely adopted standards remain undeveloped [48]. Furthermore, integrating Cyber Threat Intelligence (*CTI*) into scenario generation across platforms faces significant obstacles due to **semantic and format discrepancies**. CTI frameworks employ varying data languages and differing taxonomies (e.g., *ATT&CK* [17] tactics versus *CAPEC* [72] attack patterns), complicating automated mapping of real threat data into scenarios [14]. Addressing this issue demands significant research into interoperability frameworks, including standardized SDLs and translation/normalization layers that conform threat intelligence data with scenario orchestration.

Another critical aspect is establishing **benchmarking and comparative evaluation** frameworks to objectively assess the effectiveness of SDLs, CTI pipelines, and scenario execution platforms. Currently, evaluations tend to be qualitative and ad hoc, with **limited quantitative benchmarks** for rigorously measuring performance [5]. There is **no standardized metric** for assessing scenario expressiveness, CTI pipeline realism, or simulation fidelity. Additionally, training outcomes lack formal, standardized assessment methods, complicating objective comparison of various solutions. Establishing **cross-layer benchmarks**, such as reference scenarios or standardized attack sequences for uniform evaluation, would enable meaningful comparative analysis [30]. Such benchmarks could evaluate scenario fidelity (e.g., coverage of MITRE ATT&CK techniques), adaptability (e.g., ease of modification), and impact on trainee performance. Collaborative efforts by standards organizations or research consortia to define standardized evaluation suites and metrics are necessary. Developing these benchmarks would drive evidence-based improvements, enabling clear identification of effective innovations and enhancing overall training efficacy and cyber threat coverage in cyber range environments.

TABLE VI: Summary of key open challenges across Scenario Description Languages (SDLs), CTI-driven scenario generation pipelines, and scenario execution platforms, including cross-layer issues.

Challenge	Layer	Description
C_1 Formal Semantics and Verification	Scenario Description Languages (SDLs)	Lack of machine-checkable scenario grammars. Impedes automated reasoning and verification.
C_2 Behavioural Fidelity	Scenario Description Languages (SDLs)	Limited encoding of procedure-level attacker behavior. Requires richer ontologies or automated script generation for realism.
C_3 Trustworthiness of CTI Data	CTI-driven scenario generation pipelines	Uncertainty in threat intelligence quality and source credibility. Need methods to evaluate, filter, and use credible CTI in scenarios.
C_4 Continual Scenario Evolution	CTI-driven scenario generation pipelines	Static, snapshot-based scenario generation can not keep up with evolving threats. Need pipelines that support live updates and adapt to changing TTPs while maintaining consistency.
C_5 API-Driven Scenario Ingestion	Scenario execution platforms	Cyber ranges lack standard APIs for injecting scenarios. A unified interface is needed to enable automation and cross-platform scenario portability.
C_6 Scalable Telemetry and Observability	Scenario execution platforms	Platforms offer limited built-in instrumentation. Capturing detailed exercise telemetry (e.g. full packet capture) is hampered by performance and scaling issues, impeding analysis and feedback.
C_7 Reproducibility and Benchmarking	Cross-layer	Exercises and research results are hard to reproduce across different ranges due to non-standard scenario definitions and ad-hoc versioning. No common benchmarks or metrics exist to compare results across different range platforms [152].

E. Summary of Challenges

Our study identifies the seven key research challenges ($C_{1\text{–}7}$ detailed in the sub-sections above) that currently hamper fully integrated, CTI-driven cyber range operations across all layers. At the Scenario Description Language *(SDL)* layer, the lack of formal semantics (C_1) and limited behavior fidelity (C_2) remain major gaps. CTI-driven scenario generation pipelines face issues in ensuring trustworthy Cyber Threat Intelligence *(CTI)* origin (C_3) and in enabling continuous scenario evolution as adversaries adapt (C_4). Scenario execution platforms must overcome the lack of standardized scenario ingestion APIs (C_5) and the difficulty of providing high-fidelity telemetry at scale (C_6). Finally, a cross-layer challenge is improving reproducibility and establishing shared benchmarks (C_7) for consistent evaluation. **Table VI** summarizes these challenges, with their primary layer, and brief descriptions.

VII. Future Research Directions

Building on the seven open challenges ($C_{1\text{–}7}$) identified in ***Section VI***, we outline key avenues for future research in this section. These challenges span from fundamental issues in scenario specification to broader limitations in threat-intelligence (*CTI*) integration and platform capabilities. **Formal scenario semantics**, **behavioral fidelity**, **CTI trust**, **continuous scenario evolution**, **standardized APIs**, **high-fidelity telemetry**, and **reproducibility** represent critical gaps that truly hamper CTI-informed cyber ranges. Addressing these gaps is essential to achieve end-to-end, intelligence-driven pipelines. In particular, overcoming the fragmentation in current solutions will require focused research efforts that bridge Scenario Description Languages (*SDLs*), CTI-driven scenario generation processes, and scenario execution environments. In this section, we discuss several promising research directions aimed at resolving these open issues and enabling next-generation cyber ranges.

A. Toward Standardized Scenario Semantics

A fundamental research direction is to **formalize and standardize the semantics** of Cyber Range Training Exercise Scenario Description Languages *(CyRaTrEx SDLs or SDLs)*. As highlighted in challenge C_1 ***Formal Semantics and Verification***, most SDLs today lack machine-checkable grammars or rigorous definitions of scenario behavior. Without a formal semantic foundation, automated reasoning about scenarios—such as verifying scenario reachability or the absence of dead-ends in an attack narrative—remains infeasible. To enable robust SDL verification, future work should establish **unambiguous scenario ontologies and syntax rules** that are shared across platforms. One approach is to define a **common intermediate representation** or schema for scenarios, against which formal properties can be verified (e.g., using model-checking or constraint-solving techniques). For

instance, *Russo et al.* [16] demonstrated that encoding an SDL into logic (Datalog) permits automated consistency checks on cyber scenarios. Generalizing such methods, researchers should explore translations of scenario specifications into verification-friendly languages (e.g., *TLA*$^+$ [145], [146] or *Alloy* [147], [148]) to allow rigorous analysis of scenario logic and prerequisites. Standardized semantics, in addition to facilitating **correctness proofs** (e.g., ensuring every attack step is executable under some conditions), will also improve **interoperability** where a well-defined scenario specification could be portable between different range platforms without semantic drift.

Equally important is extending scenario semantics to capture **richer threat behavior and context**. As noted in challenge C_2 ***Behavioural Fidelity***, current SDLs often revolve around high-level tactics and techniques (e.g., *MITRE ATT&CK IDs* [17]) but omit the procedure-level details (specific commands, malware artifacts, lateral movement steps) that determine how an attack unfolds. Therefore, future research should incorporate **richer behavioral ontologies** into scenario descriptions, so that scenarios can express not just **what tactic** but **how exactly** an attacker's action is realized. Developing a community-driven ontology (or extending standards like *STIX* [57]) to represent exploits, tools, and effects consistently will enhance semantic fidelity. With more **expressive scenario models**, pipelines could automatically map real CTI feeds into concrete scenario steps, bridging the gap between threat intelligence and scenario instantiation. Hence, establishing formal and standardized scenario semantics augmented with detailed behavior modeling would provide the necessary foundation for automated scenario generation and verification across diverse cyber range platforms.

B. Toward Adaptive CTI-Driven Scenario Pipelines

An important future direction is the development of *adaptive scenario generation pipelines* driven by real-time Cyber Threat Intelligence (CTI) feeds. Unlike static scenario creation, an adaptive pipeline would **dynamically incorporate incoming intelligence updates**, addressing challenges C_3 ***Trustworthiness of CTI Data*** and C_4 ***Continual Scenario Evolution***. This approach acknowledges that cyber threats evolve continuously, and thus the scenarios used for simulation must keep pace accordingly. We have divided this future research direction into different aspects, as detailed below.

In contrast to current practices where scenarios often remain **static** snapshots, an adaptive pipeline could continuously ingest threat feed updates, such as new Indicators of Compromise *(IoCs)*, TTPs, or adversary behaviors, and update scenarios in near real-time. This would enable security teams to rapidly **refresh** attack scenarios as new threat reports emerge, ensuring relevance against the latest adversarial techniques. Recent work highlights the shortcomings of static approaches (e.g., mainstream CTI platforms excel at storing structured IoCs but struggle with unstructured intel, and manual curation is still prevalent), leading to scenario generation **latencies averaging over 48 hours** for critical vulnerabilities [83]. Such delays are not sustainable against fast-moving threats. A dynamic CTI-driven pipeline aims to achieve push-button scenario updates, where an analyst's request triggers automated scenario construction from the most recent intelligence. Eventually, this could enable on-demand, streaming scenario updates (e.g., daily/hourly), drastically reducing the window in which training scenarios lag behind real-world threats

Another major research requirement for these pipelines is robust, origin-aware CTI ingestion. Since CTI originates from diverse sources, such as open feeds, vendor reports, Information Sharing and Analysis Center *(ISACs)*, of varying credibility, the pipeline must assess and track source origin, confidence, and reliability of CTI. Ensuring the trustworthiness of ingested intelligence is critical, as unreliable or false intelligence could lead to misleading scenarios. Prior studies have underlined that CTI quality is dependent on source credibility, and automated methods are emerging to quantify this. For instance, *Yang et al.* [149] propose to evaluate CTI feeds by correlating **feed trustworthiness** with content originality and availability. Incorporating such quality assessments into the pipeline would allow filtering of low-confidence intel. The origin metadata, such as **who** reported an indicator, **when**, and through **what** chain of custody, can strengthen trust in CTI. However, practitioners note that such detailed provenance is **rarely available** in practice. Research is needed on incentive and technical mechanisms for CTI providers to supply richer origin, as well as automated validation, such as cross-correlating multiple feeds to spot inconsistencies or disinformation.

The broad vision is a largely automated, ***push-button*** system where fresh CTI can be transformed into a new scenario or updated with **minimal human intervention**. Achieving this requires filling the current automation gaps (Challenge C_4) in translating CTI into scenario narratives and technical environments. Constructing a complex attack scenario from CTI often involves **labor-intensive** steps such as manually interpreting reports, extracting relevant attacker actions, mapping them to an environment, and scripting the scenario. This process is slow and error-prone, contributing to significant lag. For example, analysts often must **manually resolve semantic discrepancies** in CTI reports, which can delay scenario builds by **days** [83]. Future research should focus on end-to-end pipelines that can automatically perform **CTI parsing**, scenario drafting, and even deployment. In a push-button paradigm, an analyst could select an intelligence report or a set of IOCs and TTPs, and the system would automatically generate a candidate scenario (including attack storyline, required infrastructure or simulators, and expected outcomes). Any new CTI that arrives (e.g., via a streaming API or TAXII feed) could trigger **incremental updates** to existing scenarios, for instance, adding a new step to reflect a discovered tactic, or modifying indicators to match the latest attacker infrastructure. This aligns with streaming cyber defense strategies, where detection content and threat hunting hypotheses are updated on the fly [157]. Similarly, scenarios in cyber ranges or training exercises would be continuously refreshed. The key research includes real-time CTI parsing algorithms, incremental scenario update techniques (to ***avoid rebuilding from scratch***), and user interfaces that allow security operators to supervise

and tweak auto-generated scenarios with a **human-in-the-loop** for quality control.

To realize the above, advanced **Artificial Intelligence *(AI)*** and knowledge representation techniques will be indispensable. As discussed before, having a standardized scenario semantics (a common ontology or schema for scenario elements) provides a **target format** for automation. The role of AI would be to bridge from unstructured or semi-structured CTI data to this scenario representation. **Natural language processing (*NLP*)**, especially using modern **Large Language Models (*LLMs*)**, can extract structured facts from free-text threat reports. Researchers have already started leveraging LLM pipelines to interpret CTI, e.g., to perform named entity and relation extraction for building knowledge graphs of threat activity. Tools like ***TTPDrill*** [158], ***TTPHunter*** [159], and ***LADDER*** [160] illustrate early progress in structuring CTI into knowledge bases of attacker tactics [75]. Likewise, the *MITRE ATT&CK* [17] framework serves as a real ontology for adversary tactics/techniques, and some frameworks map CTI reports onto ATT&CK techniques to standardize semantics. Building on such efforts, an adaptive scenario pipeline could utilize ontologies (e.g., a scenario ontology aligned with *ATT&CK* or *STIX*) to represent the narrative and technical components of an attack scenario. AI-driven mappers would translate CTI inputs into this ontology by identifying the threat actors, campaign steps, tools used, objectives, etc., and linking them into a structured scenario model. Recent prototypes underscore this potential, e.g., ***CRUcialG*** [82] automatically parses free-text CTI reports to construct Attack Scenario Graphs *(ASGs)* by extracting entities and attacker actions, then uses graph-based reasoning to verify and complete the scenario timeline. Such ontology-backed pipelines ensure that as new intelligence arrives, it can be semantically aligned with existing scenario knowledge (avoiding **semantic drift** where the scenario's story deviates from evolving attacker behaviors). Moreover, an ontology can facilitate a **union of CTI from multiple sources**, where a pipeline can merge intel about related threat actors or techniques under common scenario nodes, improving completeness and consistency. AI algorithms or simulation tools could then take the structured scenario description and instantiate it in a cyber range or test environment automatically. Hence, integrating AI for language understanding and pattern learning, with formal ontologies for scenario representation, is a promising route to achieve adaptive, CTI-driven scenario generation [75]. This will enable future systems to continuously map raw threat data into actionable simulations and exercises, significantly enhancing organizational preparedness.

C. Scenario Lifecycle Management and Reproducibility

CyRaTrEx scenarios should be treated as **evolving assets** rather than static artifacts. In current practice, once a scenario (e.g., a networked lab with an embedded attack narrative) is scripted and deployed, it often remains **unchanged** over time, even as real-world tactics and system configurations **rapidly advance**. This static approach leads to **scenario drift**, where an initially relevant exercise can become **outdated** and lose realism if not continuously updated with new threat intelligence and technology changes. Frequent refreshes are essential, e.g., recent findings show that pipelines injecting fresh CTI at least every two weeks achieved roughly **2.5× higher realism** scores than those with infrequent updates. Therefore, the motivation is to manage scenario content through its lifecycle, from creation and deployment to iterative refinement, much like software code or AI models. Treating scenarios as evolving assets acknowledges that a cyber range training exercise is never *finished*, but must be routinely enhanced to reflect the latest attacks, vulnerabilities, and defense techniques. This shift can significantly **reduce the manual effort** of reinventing scenarios from scratch, instead fostering reuse and incremental improvement. It would also enable maintaining a consistent difficulty and relevance level over time, as scenarios could gradually increase in complexity or adapt to trainees' skill growth. Hence, there is a clear need for principled **scenario lifecycle management** frameworks so that cyber range exercises remain continuously aligned with the evolving threat landscape.

A key requirement for such lifecycle management is **robust versioning and modularity** support for scenario definitions. Just as modern DevOps practices rely on source control to track code changes, scenario authors should be able to *version-control their scenarios* (e.g., via *Git*) and collaboratively improve them. By maintaining a history of scenario *builds* and changes, one can revert problematic updates, compare different scenario variants, and systematically integrate new threat intelligence (e.g., adding a recent ransomware tactic into a ransomware training scenario as version 2.0). This calls for SDLs and tools that enable a modular composition of scenarios. Modular SDL design allows common elements (network blueprints, attacker behaviors, etc.) to be reused and swapped out with minimal effort. For instance, *Russo et al.* [16] propose a **Scenario-as-Code** paradigm extending Infrastructure-as-Code *(IoC)* to design scenarios, where the corresponding *CRACK SDL* implementation is openly available [90]. In their CRACK framework, scenario building blocks (topologies, vulnerabilities, team roles) are defined as **reusable** modules, yielding high reuse rates across scenario variants (often $\sim$90% of components could be shared). Such modularity not only accelerates the creation of new exercises but also makes it easier to evolve scenarios, where an outdated component (e.g, a deprecated software or a patched CVE exploit) can be replaced in the library and seamlessly propagated to all scenarios that use it. Moreover, lifecycle frameworks should support **scenario patching**, analogous to software patching, where critical updates (e.g., injecting a newly discovered threat intel or fixing a misconfiguration) can be applied to a scenario without breaking its overall logic. Implementing these capabilities will likely require new scenario management platforms or repositories that treat scenario files, scripts, and assets as objects subjected to continuous integration pipelines and rigorous code review. In essence, the community needs to **extend DevOps principles to scenario engineering**, such as automated tests of scenario validity, style checks for SDL code, and staging environments to trial scenario updates before rollout.

Reproducibility is an equally critical pillar of scenario

lifecycle management, and one that we identify as the cross-layer challenge C_7 ***Reproducibility and Benchmarking***, affecting SDLs, CTI generation pipelines, and scenario execution platforms alike. In scientific computing, reproducibility means that an experiment can be repeated with the same results. In cyber range exercises, this translates to being able to **re-run** a scenario (or an entire training campaign) and obtain **consistent outcomes** (e.g., identical network traffic patterns or trainee performance metrics) provided the initial conditions are the same. Unfortunately, achieving this is far from trivial. Even minor, uncontrolled differences in environment configuration or timing can significantly alter what unfolds during a cyber scenario. For example, if an **identical** training scenario is executed on two different range platforms or under *slightly different* software versions, the sequence of attacker–defender interactions might **diverge**, leading to incomparable results [152]. This unpredictability undermines confidence that scenario-based findings, such as evaluations of a defensive tool or training efficacy, are sound and generalizable. To address this, future research must establish methods to **capture and preserve the exact state** of scenarios and their execution environments. Promising directions include using **containerization and virtualization snapshots** to freeze the infrastructure state, deterministic seeding of any randomized processes, and standardized initialization of threat intelligence inputs. For instance, one vision is to curate a **benchmark repository** bundling everything needed for a scenario, i.e., ***(i)*** parameterized SDL files defining the scenario logic, ***(ii)*** a snapshot of the CTI feed or threat dataset driving the scenario, and ***(iii)*** containerized images or virtual machine snapshots of the scenario execution platform configured for that scenario. By packaging scenarios with their dependencies and environment, researchers and instructors could reliably **re-deploy** a given scenario and observe the same behavior. Such scenario bundles would also facilitate ***apples-to-apples*** comparisons, where a new scenario generation algorithm or defensive technique could be evaluated on a shared baseline scenario, ensuring that differences in outcomes are due to the technique under study and not an artifact of divergent scenario conditions.

To realize scenario lifecycle management and reproducibility, several enabling technologies and practices must converge. One enabler is the integration of **DevOps-like toolchains** tailored to cyber ranges. This entails establishing continuous integration/continuous deployment *(CI/CD)* pipelines for scenarios, where whenever a scenario is updated (e.g., a new attack step is added), the pipeline would **automatically deploy the updated scenario** in a test environment, execute it (possibly in a headless or simulation mode), and verify certain invariants or learning objectives (e.g., that the attack still triggers the expected alarms and that all scoring scripts run without error). Such regression testing for scenarios would guard against unintended side-effects of updates, catching issues early just as software tests catch bugs introduced by code changes. Another enabler is **scenario version control and cataloging** infrastructure, essentially a *GitHub for scenarios*, where scenario contributions from different authors can be systematically stored, tagged (with metadata like required platform or *ATT&CK* techniques covered), and versioned. For instance, the community-driven ***REWIRE project*** [161] has outlined requirements for a scenario sharing platform that supports source control, review workflows, and CI integration. Establishing a rich public repository of versioned scenarios would also foster reproducibility where researchers could pull a specific scenario version (e.g., *RansomwareSimulation v1.2*), along with its defined environment and CTI snapshot, to reproduce a published experiment or training outcome.

D. Cyber Range Observability and Performance Feedback

Cyber ranges must not only execute realistic scenarios but also **observe and measure** everything that transpires. Integrating comprehensive telemetry streams, real-time feedback mechanisms, and **post-exercise analytics** is vital for effective training. Detailed monitoring data enables precise evaluation of participants' actions and greatly improves after-action debriefings. Likewise, **immediate feedback** during an exercise can dramatically enhance learning outcomes where trainees benefit from objective, real-time insight into their mistakes and progress, rather than waiting until the end of the exercise. Despite its importance, current platforms exhibit **significant limitations in monitoring both attacker and defender actions** and the overall system state. Many ranges provide only **partial visibility** into exercise events, often focusing on basic outcomes (e.g., scoring points or completed objectives) while neglecting fine-grained activity traces. In practice, open-source and legacy range solutions tend to offer just the *bare basics* of feedback (simple scoreboards, static hints, or rudimentary logs), without deeply instrumenting every action taken by red or blue teams. If rich telemetry is not built into the scenario environment, instructors have to resort to **manual observation and ad-hoc note-taking**, since the platform cannot automatically track or assess what unfolds. As a result, critical attacker/defender behaviors may go unrecorded. For instance, capturing full network traffic or detailed host-level events is still the exception rather than the norm (packet-level capture is present in only about ***29%*** of surveyed platforms). This is largely because of the **heavy I/O overhead and storage burden** it imposes. Furthermore, most ranges today rely on static pre-scripted difficulty and fixed scenario paths, where dynamic adjustment of scenario conditions or difficulty in response to trainee performance remains largely absent. Except for ***SCORPION*** [155] being a rare research prototype that implements on-the-fly scenario adaptation, the state of cyber range observability is fragmented and often inadequate, leaving a gap in the ability to fully monitor and understand what happens during complex exercises.

These gaps highlight the need for standardized telemetry formats, plug-and-play instrumentation, and richer evaluation dashboards in future range designs in accordance with the identified challenge C_6 ***Scalable Telemetry and Observability***. A common telemetry schema, and open APIs to export/import it, would allow different tools and platforms to **share** exercise data seamlessly. In current practice, each cyber range tends to implement its own ad-hoc logging and scoring format, making it **difficult** to aggregate data or **integrate third-party**

analytics. Therefore, establishing community standards should be identified as a priority to decouple scenario instrumentation from any single platform's implementation. With standard interfaces, one could easily attach new monitoring probes or data collectors (plug-and-play instrumentation) into any scenario without custom engineering, and combine telemetry from **multiple** sources into a unified timeline of the exercise. Equally important is **presenting the collected data** in meaningful ways. **User-friendly dashboards** and analytic tools should transform raw log data into actionable training insights. Advanced commercial ranges already demonstrate the value of this approach, e.g., platforms like ***Cyberbit*** [24] and ***RangeForce*** [25] automatically measure individual and team performance metrics, such as detection rates and response times, and display them on comprehensive dashboards to pinpoint skill gaps. By adopting similar evaluation interfaces, next-generation cyber ranges can provide instructors and trainees with clear, **real-time indicators** of how well defensive actions are working, how quickly attacks are detected, and where improvements are needed.

Increased **observability** involves more than just scoring an exercise. It can fundamentally improve training effectiveness and enable new adaptive capabilities. Rich telemetry and performance feedback loops allow a cyber range to move beyond universal scenarios toward **adaptive** training experiences. For example, ***SCORPION*** [155] cyber range monitors a wide array of participant behaviors, including biometric signals like **heart rate**, and uses those inputs to **dynamically adjust scenario difficulty** and guidance in real time. If a trainee is struggling with a challenge, the system might lower the complexity or provide a timely hint. On the other hand, if a trainee is excelling, the range can ***introduce new hurdles to keep them engaged***. Such adaptive scenario control, driven by continuous observability, personalizing the exercise to each learner's skill level and stress tolerance, is shown to boost engagement and skill acquisition. Even in non-adaptive settings, **high-fidelity** logs and analytics help instructors pinpoint exactly **how** a breach occurred or **why** a defense failed, enabling **targeted feedback** that strengthens the trainee's understanding. Thus, observability is a prerequisite for treating cyber ranges not just as technical sandboxes but as data-driven learning environments. Moreover, a standardized and thorough observability layer can facilitate **systematic benchmarking and research** on cyber defense exercises themselves. If multiple range platforms adhere to common telemetry formats and capture comparable performance data, the community can begin to quantitatively compare **what works** in cyber training across different environments. For instance, one could measure how varying the volume of background traffic or the timing of attacker moves affects detection rates and defender fatigue, provided all those aspects are logged consistently.

Finally, realizing integrated observability and feedback in cyber ranges raises several open research challenges. One challenge is improving **logging fidelity without overwhelming the system** or analysts with data. Capturing every packet, keystroke, and system call in a large exercise can quickly produce data in the order of tens of gigabytes per hour in a moderate-sized lab, straining storage and computational resources. Novel techniques for **selective and intelligent logging** are needed, such as using kernel-level filters with **Extended Berkeley Packet Filter** *(eBPF) [162]*, to record only suspicious events or compress and summarize data in real time. A related issue is minimizing the performance **overhead** of instrumentation. Monitoring agents and network taps should have a minimal footprint, so as not to alter the realism of the exercise or introduce detectable lag. Research is required to design **efficient, low-latency telemetry pipelines** that still preserve crucial detail. Another key consideration is **privacy-preserving instrumentation**. By its nature, extensive cyber range logging may capture **sensitive information** about participants' actions, strategies, or even physiological responses. For example, recording biometric stress indicators or detailed keystroke logs can provide valuable training data, but also raise **privacy and ethical concerns**. Future platforms should incorporate safeguards such as **anonymization** of personal data, access controls for sensitive logs, and clear **consent** from participants on what data is collected. Thus, enhancing cyber range observability and feedback will require balancing the richness of data with practical constraints. This research direction is crucial for evolving cyber ranges into fully instrumented, intelligent training grounds *where every attack and defense is measured, learned from, and continuously improved upon*.

VIII. Conclusion

This survey was motivated by the need to keep cyber range training aligned with a rapidly evolving threat landscape. Modern cyber ranges must support exercises of **real-world complexity** with high realism, yet designing such scenarios manually is **costly** and **error-prone**. We have systematically reviewed existing approaches to threat-informed scenario generation and execution, highlighting the roles of formal Scenario Definition Languages *(SDLs)*, Cyber Threat Intelligence *(CTI)* sources, and CyRaTrEx scenario generation and execution pipelines. In particular, we summarized how SDLs based on open standards, such as *OASIS TOSCA* [45], enable declarative specification of scenario elements, and how CTI frameworks, such as *MITRE ATT&CK* [17], provide **adversarial tactics and techniques** that should populate these scenarios. Our contribution is a holistic mapping of an end-to-end pipeline linking threat intelligence, scenario design, generation, and execution, along with an identification of key gaps in each stage.

We reiterate that an **end-to-end threat-informed** pipeline is essential for effective cyber range training. By continuously integrating current threat data into scenario design, exercises remain relevant to emerging adversary behaviors. We emphasized examples of automated pipelines, such as the *CRACK* [16] framework, that connect formal design, verification, and deployment of scenarios. Such pipelines can **reduce manual errors** and enable more frequent, diverse exercises. This is critical as only a few organizations can currently afford high-fidelity live-fire training. Thus, our survey shows that closing the loop from CTI ingestion through scenario generation to execution is vital for scaling cyber training.

Beyond human training, researchers have also introduced gymnasium-like environments, such as *CybORG* [163], for developing autonomous cyber agents within simulated range scenarios, hinting at future integration of AI-driven cyber defense training.

A central theme of our work is bridging gaps across SDLs, CTI integration in scenario generation, and scenario execution. We demonstrated that SDLs can serve as an interface between abstract threat models and concrete deployments. For example, *TOSCA*-based SDLs allow **declarative encoding** of network topologies, vulnerabilities, and objectives, and in principle can be **extended** to include threat intelligence, for instance, by tagging nodes with *CAPEC* [72]/*CVE* [61] references. We reported prior work that explicitly incorporates CTI into scenario generation as a multi-step process, e.g., **mapping threat information** into attack sequences. In our survey, we illustrated how automated tools can consume SDL specifications and instantiate them in a cyber range. Nonetheless, challenges remain, such as **standard representations** for dynamic threat feeds and end-to-end verification of deployed scenarios against intended threat models.

Looking ahead, we argue that realizing this vision demands **community-wide standardization** and **collaboration**. Standardized scenario description formats built on accepted standards, such as *TOSCA* [45], and shared CTI schemas, such as *STIX/TAXII* [57], [58], *MITRE ATT&CK* [17] taxonomies, can be key to interoperability. Collaborative efforts, such as industry consortia or research centers, must **codify best practices and open data**. For instance, *EU* initiatives like *FORESIGHT* [164] and *ECHO* [165] aim to federate cyber ranges and develop innovative training curricula across organizations. The *EU REWIRE alliance* [108], [161], [166] has likewise delivered a cyber range establishment **methodology**, a unified scenario development **framework**, and policy **recommendations** for cyber ranges, providing a comprehensive **blueprint** for future platforms. We encourage the formation of **cross-organizational** platforms to share scenario libraries, tooling, and lessons learned, e.g., a recent federation of sector-specific cyber ranges demonstrated both the potential and challenges of linking disparate environments [167]. With broad adoption of threat-informed standards and open collaboration, the community can continuously refine and scale the cyber training pipeline, greatly improving cyber readiness across industries. Moreover, it is crucial that industry and academia partner to establish new cyber ranges (e.g., a joint *Nortal–Mastercard* cyber resilience center [168]), underscoring the value of cross-organizational cooperation, and more importantly, to offer managed breach-and-attack simulation services to continually assess organizational defenses.

REFERENCES

[1] R. Petersen, D. Santos, K. Wetzel, M. Smith, and G. Witte, "Workforce Framework for Cybersecurity (NICE Framework)," National Institute of Standards and Technology, Tech. Rep. NIST Special Publication (SP) 800-181 Rev. 1, Nov. 2020. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/181/r1/final

[2] K. Yonemura, J. Sato, R. Komura, and M. Matsuoka, "Practical security education on combination of OT and ICT using gamfication method," in *2018 IEEE Global Engineering Education Conference, EDUCON 2018, Santa Cruz de Tenerife, Tenerife, Islas Canarias, Spain, April 17-20, 2018*. IEEE, 2018, pp. 746–750. [Online]. Available: https://doi.org/10.1109/EDUCON.2018.8363305

[3] C. Magazine, "Cybersecurity Jobs Report: 3.5 Million Unfilled Positions In 2025," Feb. 2018. [Online]. Available: https://cybersecurityventures.com/jobs/

[4] George Hatzivasilis, Kostas Fysarakis, and Sotiris Ioannidis, "Cyber-Ranges as a Mean of Security Culture Establishment," 2020. [Online]. Available: https://ercim-news.ercim.eu/en121/r-i/cyber-ranges-as-a-mean-of-security-culture-establishment

[5] T. Balon and I. A. Baggili, "Cybercompetitions: A survey of competitions, tools, and systems to support cybersecurity education," *Education and Information Technologies*, vol. 28, no. 9, pp. 11 759–11 791, Feb. 2023. [Online]. Available: https://doi.org/10.1007/s10639-022-11451-4

[6] "Facebook CTF | Menlo Park CA." [Online]. Available: https://www.facebook.com/officialctf

[7] K. C. . C. LLC, "CTFd : The Easiest Capture The Flag Framework." [Online]. Available: https://ctfd.io/

[8] "Cyber Coalition." [Online]. Available: https://www.act.nato.int/activities/cyber-coalition/

[9] "Locked Shields." [Online]. Available: https://ccdcoe.org/locked-shields/

[10] "Crossed Swords." [Online]. Available: https://ccdcoe.org/exercises/crossed-swords/

[11] CyberDefenders, "What is a Cyber Range?" [Online]. Available: https://cyberdefenders.org/blog/cyber-range/

[12] Michael VanPutte, "The Future of Cyber Experimentation and Testing," 2009. [Online]. Available: https://www.usenix.org/legacy/event/cset09/tech/slides/vanputte.pdf

[13] A. Shehu and R. Kushe, "A Cyber Attack Scenario Using SSFNet," in *2011 14th International Conference on Network-Based Information Systems*, Sep. 2011, pp. 690–693, iSSN: 2157-0426. [Online]. Available: https://ieeexplore.ieee.org/document/6041584

[14] E. Russo, G. Costa, and A. Armando, *Scenario Design and Validation for Next Generation Cyber Ranges*, Nov. 2018, pages: 4.

[15] G. Costa, E. Russo, and A. Armando, "Automating the Generation of Cyber Range Virtual Scenarios with VSDL," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 13, no. 4, pp. 61–80, Dec. 2022. [Online]. Available: https://jowua.com/wp-content/uploads/2023/01/jowua-v13n4-4.pdf

[16] E. Russo, G. Costa, and A. Armando, "Building next generation Cyber Ranges with CRACK," *Computers & Security*, vol. Volume 95, Apr. 2020.

[17] "MITRE ATT&CK®." [Online]. Available: https://attack.mitre.org/

[18] A. Zacharis and C. Patsakis, "AiCEF: an AI-assisted cyber exercise content generation framework using named entity recognition," *International Journal of Information Security*, vol. 22, no. 5, pp. 1333–1354, Oct. 2023, company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 5 Publisher: Springer Berlin Heidelberg. [Online]. Available: https://link-springer-com.proxy.bnl.lu/article/10.1007/s10207-023-00693-z

[19] "OpenBAS-Platform/openbas," Jun. 2025, original-date: 2016-09-12T23:36:20Z. [Online]. Available: https://github.com/OpenBAS-Platform/openbas

[20] Mark Bradbury, "Cyber Operations, Development and Evaluation (CODE) Center," 2017. [Online]. Available: https://itea.org/images/pdf/conferences/2017_Cyber/Proceedings/Bradbury%20-%20CODE%20Center.pdf

[21] ˜Cappetta, "AWS Cyber Range— The Ultimate Cyber Lab Overview," Jun. 2020. [Online]. Available: https://medium.com/aws-cyber-range/aws-cyber-range-the-ultimate-cyber-lab-overview-3affcca1c842

[22] "Tailored Cyber Range Solutions | SimSpace," Jul. 2024. [Online]. Available: https://simspace.com/

[23] "X-Force Cyber Range | IBM." [Online]. Available: https://www.ibm.com/services/xforce-cyber-range

[24] "Cyberbit." [Online]. Available: https://www.cyberbit.com/platform/cyber-range/

[25] "RangeForce Cloud-Based Cyber Range | Cybersecurity Training." [Online]. Available: https://www.rangeforce.com

[26] Keysight, "Cyber Range Solution," section: Article Section. [Online]. Available: https://www.keysight.com/us/en/products/network-test/cyber-range-services.html

[27] J. Davis and S. Magrath, "A survey of cyber ranges and testbeds," 2013.

[28] T. Benzel, "The science of cyber security experimentation: the DETER project," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC '11. New York, NY, USA: Association for Computing Machinery, Dec. 2011, pp. 137–148. [Online]. Available: https://dl.acm.org/doi/10.1145/2076732.2076752

[29] M. M. Yamin, B. Katt, and V. Gkioulos, "Cyber ranges and security testbeds: Scenarios, functions, tools and architecture," *Computers & Security*, vol. 88, p. 101636, Jan. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404819301804

[30] 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, p. 1809, Feb. 2021.

[31] P. Lillemets, N. Bashir Jawad, J. Kashi, A. Sabah, and N. Dragoni, "A Systematic Review of Cyber Range Taxonomies: Trends, Gaps, and a Proposed Taxonomy," *Future Internet*, vol. 17, no. 6, p. 259, Jun. 2025, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1999-5903/17/6/259

[32] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *bmj*, vol. 372, 2021.

[33] 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," Oct. 2020, arXiv:2010.06850 [cs]. [Online]. Available: http://arxiv.org/abs/2010.06850

[34] 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, Jul. 2024, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1999-5903/16/7/231

[35] C. Steininger, L. Götz, and M. Schopp, "How accessible is cybersecurity training? A survey on the accessibility, capabilities, and technology stack of Cyber Ranges," Apr. 2025. [Online]. Available: https://www.techrxiv.org/users/908351/articles/1283135-how-accessible-is-cybersecurity-training-a-survey-on-the-accessibility-capabilities-and-technology-stack-of-cyber-ranges?commit=0cdd8eaa589a451e58c47ef6d1eba04b971f663f

[36] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, "MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*. Vienna Austria: ACM, Oct. 2016, pp. 49–56. [Online]. Available: https://dl.acm.org/doi/10.1145/2994539.2994542

[37] C. J. Ihrig, "JavaScript Object Notation," in *Pro Node.js for Developers*. Apress, Berkeley, CA, 2013, pp. 263–270. [Online]. Available: https://link-springer-com.proxy.bnl.lu/chapter/10.1007/978-1-4302-5861-2_17

[38] V. Sinha, F. Doucet, C. Siska, R. Gupta, S. Liao, and A. Ghosh, "YAML: a tool for hardware design visualization and capture," in *Proceedings 13th International Symposium on System Synthesis*, Sep. 2000, pp. 9–14, iSSN: 1080-1820. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/874023

[39] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)."

[40] G. Costa, E. Russo, and A. Armando, *Automating the Generation of Cyber Range Virtual Scenarios with VSDL*, Jan. 2020.

[41] M. Shirmohammadi, "CST-SDL: A Scenario Description Language for Collaborative Security Training in Cyber Ranges."

[42] P. Martou, K. Mens, B. Duhoux, and A. Legay, "Generating Virtual Scenarios for Cyber Ranges from Feature-Based Context-Oriented Models: A Case Study," in *Proceedings of the 14th ACM International Workshop on Context-Oriented Programming and Advanced Modularity*, 2022. [Online]. Available: https://dial.uclouvain.be/pr/boreal/object/boreal:263359

[43] Y. Yamato, M. Muroi, K. Tanaka, and M. Uchimura, "Development of template management technology for easy deployment of virtual resources on openstack," *J. Cloud Comput.*, vol. 3, p. 7, 2014. [Online]. Available: https://doi.org/10.1186/s13677-014-0007-3

[44] P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorný, and M. Kudlacek, "Unleashing full potential of ansible framework: University labs administration," in *22nd Conference of Open Innovations Association, FRUCT 2018, Jyväskylä, Finland, May 15-18, 2018*. IEEE, 2018, pp. 144–150. [Online]. Available: https://doi.org/10.23919/FRUCT.2018.8468270

[45] S. Moser, P. Lipton, T. Spatzier, and D. Palma, "TOSCA Topology and Orchestration Specification for Cloud Applications Version 1.0," Nov. 2013.

[46] C. Barrett and C. Tinelli, "Satisfiability modulo theories," in *Handbook of model checking*. Springer, 2018, pp. 305–343.

[47] A. D. Costa and J. Kuusijarvi, "Programmatic Description Language for Cyber Range Topology Creation," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. Genoa, Italy: IEEE, Jun. 2022, pp. 403–412. [Online]. Available: https://ieeexplore.ieee.org/document/9799394/

[48] G. Hatzivasilis, S. Ioannidis, M. Smyrlis, G. Spanoudakis, F. Frati, C. Braghin, E. Damiani, H. Koshutanski, G. Tsakirakis, T. Hildebrandt, L. Goeke, S. Pape, O. Blinder, M. Vinov, G. Leftheriotis, M. Kunc, F. Oikonomou, G. Magilo, V. Petrarolo, and R. Bordianu, *The THREAT-ARREST Cyber Range Platform*, Jul. 2021, pages: 427. [Online]. Available: http://dx.doi.org/10.1109/CSR51186.2021.9527963

[49] Z. C. Schreuders, T. Shaw, M. Shan-A-Khuda, G. Ravichandran, J. Keighley, and M. Ordean, "Security Scenario Generator (secgen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events," 2017. [Online]. Available: https://www.usenix.org/conference/ase17/workshop-program/presentation/schreuders

[50] M. University, "KYPO Cyber Range Platform." [Online]. Available: https://crp.kypo.muni.cz/

[51] "CONCORDIA releases an open-source Cyber Range platform!" [Online]. Available: https://www.concordia-h2020.eu/news/concordia-releases-an-open-source-cyber-range-platform/

[52] A. Hussain, Y. Pradkin, and J. Heidemann, "Replay of malicious traffic in network testbeds," in *2013 IEEE International Conference on Technologies for Homeland Security (HST)*. Waltham, MA, USA: IEEE, Nov. 2013, pp. 322–327. [Online]. Available: http://ieeexplore.ieee.org/document/6699022/

[53] B. Ferguson, A. Tall, and D. Olsen, "National Cyber Range Overview," in *2014 IEEE Military Communications Conference*, Oct. 2014, pp. 123–128, iSSN: 2155-7586. [Online]. Available: https://ieeexplore.ieee.org/document/6956748

[54] M. D. R. Team., "Cyberbattlesim," https://github.com/microsoft/cyberbattlesim, 2021, created by Christian Seifert, Michael Betser, William Blum, James Bono, Kate Farris, Emily Goren, Justin Grana, Kristian Holsheimer, Brandon Marken, Joshua Neil, Nicole Nichols, Jugal Parikh, Haoran Wei.

[55] J. E. Hannay, A. Stolpe, and M. M. Yamin, "Toward AI-Based Scenario Management for Cyber Range Training," in *HCI International 2021 - Late Breaking Papers: Multimodality, eXtended Reality, and Artificial Intelligence*, C. Stephanidis, M. Kurosu, J. Y. C. Chen, G. Fragomeni, N. Streitz, S. Konomi, H. Degen, and S. Ntoa, Eds. Cham: Springer International Publishing, 2021, vol. 13095, pp. 423–436, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-030-90963-5_32

[56] "CyberRange: integration and simulation solution | Airbus," Jun. 2024. [Online]. Available: https://cyber.airbus.com/en/products/cyberrange

[57] "About STIX | STIX Project Documentation." [Online]. Available: https://stixproject.github.io/about/

[58] "Introduction to TAXII." [Online]. Available: https://oasis-open.github.io/cti-documentation/taxii/intro

[59] I. T. L. Computer Security Division, "Security Content Automation Protocol Version 2 (SCAP v2) | CSRC | CSRC," Dec. 2018. [Online]. Available: https://csrc.nist.gov/projects/security-content-automation-protocol-v2/faqs

[60] "National Institute of Standards and Technology," Aug. 2025, last Modified: 2025-09-08T10:16-04:00. [Online]. Available: https://www.nist.gov/

[61] "CVE: Common Vulnerabilities and Exposures." [Online]. Available: https://www.cve.org/

[62] A. Zacharis, V. Katos, and C. Patsakis, "Integrating AI-driven threat intelligence and forecasting in the cyber security exercise content generation lifecycle," *International Journal of Information Security*, vol. 23, no. 4, pp. 2691–2710, Aug. 2024. [Online]. Available: https://doi.org/10.1007/s10207-024-00860-w

[63] "CWE - Common Weakness Enumeration." [Online]. Available: https://cwe.mitre.org/

[64] "MITRE D3FEND Knowledge Graph." [Online]. Available: https://d3fend.mitre.org/

[65] P. E. Kaloroumakis and M. J. Smith, "Toward a Knowledge Graph of Cybersecurity Countermeasures."

[66] S. Radack, "ITL BULLETIN FOR SEPTEMBER 2010."

[67] I. T. L. Computer Security Division, "Security Content Automation Protocol | CSRC | CSRC," Dec. 2016. [Online]. Available: https://csrc.nist.gov/projects/security-content-automation-protocol

[68] AlienVault, "LevelBlue - Open Threat Exchange." [Online]. Available: https://otx.alienvault.com/

[69] "What is OTX?" [Online]. Available: https://cybersecurity.att.com/documentation/usm-appliance/otx/about-otx.htm?Highlight=AlienVault%20Open%20Threat%20Exchange

[70] "The VERIS Framework." [Online]. Available: https://verisframework.org/

[71] N. C. W. Framework, "Nice cybersecurity workforce framework (ncwf): Draft nist special publication 800-181."

[72] "CAPEC - About CAPEC." [Online]. Available: https://capec.mitre.org/about/index.html

[73] S. Barnum, "Common attack pattern enumeration and classification (capec) schema," *Department of Homeland Security*, 2008.

[74] S. Barnum and A. Sethi, "Attack Patterns as a Knowledge Resource for Building Secure Software."

[75] R. Fieblinger, M. T. Alam, and N. Rastogi, "Actionable Cyber Threat Intelligence Using Knowledge Graphs and Large Language Models." IEEE Computer Society, Jul. 2024, pp. 100–111. [Online]. Available: https://www.computer.org/csdl/proceedings-article/eurospw/2024/672900a100/1ZyWMlXUji8

[76] J. Wynn, J. Whitmore, W. Coconato, and S. McCracken, "Critical infrastructure cyberspace analysis tool (cicat): Capability description," 2020.

[77] J. Wynn, "Critical Infrastructure Cyberspace Analysis Tool (CICAT) Capability Description."

[78] I. G. P. Eryawana, G. M. A. Sasmitaa, and A. K. A. Cahyawan, "Nist sp 800-30."

[79] I. Song, S. Jeon, D. Kim, M. G. Lee, and J. T. Seo, "GENICS: A Framework for Generating Attack Scenarios for Cybersecurity Exercises on Industrial Control Systems," *Applied Sciences*, vol. 14, no. 2, p. 768, Jan. 2024, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/14/2/768

[80] J. Franklin, C. Wergin, H. Booth *et al.*, "Cvss implementation guidance," *National Institute of Standards and Technology, NISTIR-7946*, 2014.

[81] J. Meier, *Improving web application security: threats and countermeasures*. O'Reilly Media, Inc., 2003.

[82] W. Cheng, T. Zhu, T. Chen, Q. Yuan, J. Ying, H. Li, C. Xiong, M. Li, M. Lv, and Y. Chen, "CRUcialG: Reconstruct Integrated Attack Scenario Graphs by Cyber Threat Intelligence Reports," Oct. 2024, arXiv:2410.11209 [cs]. [Online]. Available: http://arxiv.org/abs/2410.11209

[83] J. Qian, W. Shang, L. Xu, J. Luo, Z. Li, and Y. Chen, "Toward Reproducible Cyberattack Reconstruction: A Semi-Automated Framework Leveraging Standardized CTI Reports," May 2025. [Online]. Available: https://www.preprints.org/manuscript/202505.0478/v1

[84] A. Saha, J. Blasco, L. Cavallaro, and M. Lindorfer, "ADAPT it! Automating APT Campaign and Group Attribution by Leveraging and Linking Heterogeneous Files," in *The 27th International Symposium on Research in Attacks, Intrusions and Defenses*. Padua Italy: ACM, Sep. 2024, pp. 114–129. [Online]. Available: https://dl.acm.org/doi/10.1145/3678890.3678909

[85] A. Syed, B. Nour, M. Pourzandi, C. Assi, and M. Debbabi, "Comprehensive Advanced Persistent Threats Dataset," *IEEE Networking Letters*, pp. 1–1, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10929738/

[86] A. Syed, "AbSamad99/APTsDataset," Mar. 2025, original-date: 2025-03-13T15:52:33Z. [Online]. Available: https://github.com/AbSamad99/APTsDataset

[87] "Advanced Persistent Threats (APTs) campaigns database." [Online]. Available: https://research.vu.nl/en/datasets/advanced-persistent-threats-apts-campaigns-database

[88] E. WG5, "Understanding Cyber Ranges: From Hype to Reality," ESCO, Tech. Rep., 2020. [Online]. Available: https://ecs-org.eu/ecso-uploads/2023/05/2020_SWG-5.1_paper_UnderstandingCyberRanges_final_v1.0-update.pdf

[89] N. E. Kolokotronis, "FORESIGHT project: Development of a federated cyber range platform and innovative training curricula," 2022. [Online]. Available: https://www.cyber-mar.eu/wp-content/uploads/2022/01/CYBER-MAR_QMR_9_WMU_20DEC21.pdf

[90] enricorusso, "enricorusso/CRACK," Dec. 2024, original-date: 2019-09-29T11:23:25Z. [Online]. Available: https://github.com/enricorusso/CRACK

[91] T. Gustafsson and J. Almroth, "Cyber Range Automation Overview with a Case Study of CRATE," in *Secure IT Systems*, ser. LCSB, M. Asplund and S. Nadjm-Tehrani, Eds., vol. 12556. Cham: Springer International Publishing, 2021, pp. 192–209. [Online]. Available: https://doi.org/10.1007/978-3-030-70852-8_12

[92] K. Shirinkin, *Getting Started with Terraform*. Packt Publishing Ltd, 2017.

[93] MITRE Corporation, "MITRE Caldera: GitHub," Feb. 2024, original-date: 2017-11-29T01:25:10Z. [Online]. Available: https://github.com/mitre/caldera

[94] "MITRE CALDERA: A Scalable, Automated Adversary Emulation Platform." [Online]. Available: https://caldera.mitre.org/

[95] "Operationalize Threat Intel | Cyber Range Defense Tactics," Dec. 2024, section: Cyber Range Technology. [Online]. Available: https://simspace.com/blog/turning-insights-into-action-operationalizing-threat-intelligence-with-cyber-ranges/

[96] "Talos IR - Cyber Range Training || Cisco Talos Intelligence Group - Comprehensive Threat Intelligence." [Online]. Available: https://talosintelligence.com/incident_response/cyberrange

[97] "Gamified cybersecurity training solutions by Circadence." [Online]. Available: https://circadence.com/

[98] E. WG5, "Cyber Range Features Checklist & List of European Providers," ESCO, Tech. Rep. 2025 edition, 2025. [Online]. Available: https://ecs-org.eu/ecso-uploads/2025/02/Cyber-Range-Features-Checklist-List-of-European-Providers-2.pdf

[99] V. Giuliano and V. Formicola, "ICSrange: A Simulation-based Cyber Range Platform for Industrial Control Systems," 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1909.01910

[100] A. Dehlaghi, "ICSSIM," Jun. 2025, original-date: 2022-04-22T08:52:12Z. [Online]. Available: https://github.com/AlirezaDehlaghi/ICSSIM

[101] S. Hankewitz, "Estonian companies help set up a space cyber range," Nov. 2023. [Online]. Available: https://estonianworld.com/technology/estonian-companies-help-set-up-a-space-cyber-range/

[102] "CR14 Homepage." [Online]. Available: https://www.cr14.ee/

[103] "CRATE – Sweden's national cyber training facility." [Online]. Available: https://www.foi.se/en/foi/research/information-security/crate---swedens-national-cyber-training-facility.html

[104] 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," in *Proceedings of the European Interdisciplinary Cybersecurity Conference*. Rennes France: ACM, Nov. 2020, pp. 1–6. [Online]. Available: https://dl.acm.org/doi/10.1145/3424954.3424959

[105] "AIT CyberRange." [Online]. Available: https://cyberrange.at/

[106] "Perforce Puppet: Infrastructure Automation & Operations at Scale." [Online]. Available: https://www.puppet.com/

[107] "kubernetes/kubernetes: Production-Grade Container Scheduling and Management." [Online]. Available: https://github.com/kubernetes/kubernetes

[108] REWIRE Cybersecurity Alliance, "Cyber Range Establishment methodology and roadmap," Tech. Rep., 2022. [Online]. Available: https://rewireproject.eu/wp-content/uploads/2022/12/R4.1.1-Cyberrange-Establishment-methodology-and-roadmap_vFINAL.pdf

[109] "What is KVM?" [Online]. Available: https://www.redhat.com/en/topics/virtualization/what-is-KVM

[110] "VMware by Broadcom - Cloud Computing for the Enterprise." [Online]. Available: https://www.vmware.com

[111] "Docker: Accelerated Container Application Development." [Online]. Available: https://www.docker.com/

[112] "Podman." [Online]. Available: https://podman.io/

[113] TryCatchHCF, "DumpsterFire - GitHub," May 2025, original-date: 2017-10-05T23:44:54Z. [Online]. Available: https://github.com/TryCatchHCF/DumpsterFire

[114] D. Kim, S. Jeon, K. Kim, J. Kang, S. Lee, and J. T. Seo, "Guide to developing case-based attack scenarios and establishing defense strategies for cybersecurity exercise in ICS environment," *The Journal of Supercomputing*, vol. 80, no. 15, pp. 21 642–21 675, Oct. 2024. [Online]. Available: https://doi.org/10.1007/s11227-024-06273-9

[115] Y. Fu, W. Han, and D. Yuan, "Disentangled Orchestration on Cyber Ranges," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 04, pp. 2344–2360, Jul. 2024, publisher: IEEE Computer Society. [Online]. Available: https://www.computer.org/csdl/journal/tq/2024/04/10214377/1PuPcasBELC

[116] S. Gadirli, "System Monitor — Sysmon," Jul. 2023. [Online]. Available: https://medium.com/@sh.g/system-monitor-sysmon-4f9f5ff9267b

[117] "Network Security Monitoring with Zeek." [Online]. Available: https://www.pluralsight.com/paths/network-security-monitoring-with-zeek

[118] SPARTA, "D9.4 Pilot of Cyber training & exercise Framework," 2021. [Online]. Available: https://www.sparta.eu/assets/deliverables/SPARTA-D9.4-Pilot-of-Cyber-training-exercise-Framework-PU-M30.pdf

[119] A. Rossini, K. Kritikos, N. Nikolov, J. Domaschka, F. Griesinger, D. Seybold, D. Romero, M. Orzechowski, G. Kapitsaki, and A. Achilleos, *The Cloud Application Modelling and Execution Language (CAMEL)*, Mar. 2017.

[120] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. Pearson Education, 2008.

[121] "EDURange | Welcome." [Online]. Available: https://edurange.org/index.html

[122] R. Weiss, F. Turbak, J. Mache, and M. E. Locasto, "Cybersecurity education and assessment in edurange," *IEEE Security & Privacy*, vol. 15, no. 03, pp. 90–95, 2017.

[123] G. Bernardinetti, S. Iafrate, and G. Bianchi, "Nautilus: A Tool For Automated Deployment And Sharing Of Cyber Range Scenarios," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ser. ARES '21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 1–7. [Online]. Available: https://dl.acm.org/doi/10.1145/3465481.3469182

[124] C. Pham, D. Tang, K.-i. Chinen, and R. Beuran, "CyRIS: a cyber range instantiation system for facilitating security training," in *Proceedings of the 7th Symposium on Information and Communication Technology*, ser. SoICT '16. New York, NY, USA: Association for Computing Machinery, Dec. 2016, pp. 251–258. [Online]. Available: https://dl.acm.org/doi/10.1145/3011077.3011087

[125] M. Lupinacci, F. Blefari, F. Romeo, F. A. Pironti, and A. Furfaro, "ARCeR: an Agentic RAG for the Automated Definition of Cyber Ranges," Apr. 2025, arXiv:2504.12143 [cs]. [Online]. Available: http://arxiv.org/abs/2504.12143

[126] "OpenCTI-Platform/opencti," May 2025, original-date: 2018-12-17T22:57:34Z. [Online]. Available: https://github.com/OpenCTI-Platform/opencti

[127] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[128] "Emulab : Flux Research Group." [Online]. Available: https://www.flux.utah.edu/project/emulab

[129] "BAS+ | SCYTHE." [Online]. Available: https://scythe.io/services/bas

[130] "Cyber Ranges | SANS Institute." [Online]. Available: https://www.sans.org/cyber-ranges/

[131] "Cyber Range & Cyber Security Solutions - DIATEAM." [Online]. Available: https://www.diateam.net/

[132] C. Technologies, "Cyber Range Solutions | SaaS & Bespoke On-prem | CybExer." [Online]. Available: https://cybexer.com

[133] "Services - Cyber Resilience." [Online]. Available: https://nortal.com/services/cyber-resilience/

[134] "CITEF - Powering RHEA Group's Next Generation Cyber-Range Service," 2021. [Online]. Available: https://www.rheagroup.com/wp-content/uploads/2021/09/rhea-group-citef-factsheet-english-sept21.pdf

[135] "Metta," Jun. 2025, original-date: 2017-11-01T21:24:47Z. [Online]. Available: https://github.com/uber-common/metta

[136] "A Data Driven Comparison of Open Source Adversary Emulation Tools." [Online]. Available: https://www.picussecurity.com/resource/blog/data-driven-comparison-between-open-source-adversary-emulation-tools

[137] "Infection Monkey - GitHub," May 2025, original-date: 2015-08-30T07:22:51Z. [Online]. Available: https://github.com/guardicore/monkey

[138] endgameinc, "Red Team Automation (RTA) - GitHub," Jun. 2025, original-date: 2018-03-19T19:59:39Z. [Online]. Available: https://github.com/endgameinc/RTA

[139] "Stratus Red Team." [Online]. Available: https://stratus-red-team.cloud/

[140] "Akamai Infection Monkey." [Online]. Available: https://www.akamai.com/infectionmonkey

[141] "Cloud Computing Services - Amazon Web Services (AWS)." [Online]. Available: https://aws.amazon.com/

[142] "Cloud Computing Services | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-us

[143] "What is Google Cloud Platform (GCP)?" [Online]. Available: https://www.pluralsight.com/resources/blog/cloud/what-is-google-cloud-platform-gcp

[144] Keysight, "BreakingPoint," section: Article Section. [Online]. Available: https://www.keysight.com/us/en/products/network-security/breakingpoint.html

[145] L. Lamport, "Specifying Concurrent Systems with TLA+."

[146] "TLA+." [Online]. Available: https://github.com/tlaplus

[147] "alloytools.org." [Online]. Available: https://alloytools.org/

[148] "Practical Alloy." [Online]. Available: https://practicalalloy.github.io/

[149] L. Yang, M. Wang, and W. Lou, "An automated dynamic quality assessment method for cyber threat intelligence," *Computers & Security*, vol. 148, p. 104079, Sep. 2024.

[150] M. R. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "What are the attackers doing now? Automating cyber threat intelligence extraction from text on pace with the changing threat landscape: A survey," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–36, Dec. 2023, arXiv:2109.06808 [cs]. [Online]. Available: http://arxiv.org/abs/2109.06808

[151] L. Alevizos and M. Dekker, "Towards an AI-Enhanced Cyber Threat Intelligence Processing Pipeline," *Electronics*, vol. 13, no. 11, p. 2021, May 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/11/2021

[152] T. Tarman, T. Rollins, L. Swiler, J. Cruz, E. Vugrin, H. Huang, A. Sahu, P. Wlazlo, A. Goulart, and K. Davis, "Comparing reproduced cyber experimentation studies across different emulation testbeds," in *Proceedings of the 14th Cyber Security Experimentation and Test Workshop*, ser. CSET '21. New York, NY, USA: Association for Computing Machinery, Sep. 2021, pp. 63–71. [Online]. Available: https://dl.acm.org/doi/10.1145/3474718.3474725

[153] "https://www.nexovagroup.eu/sites/default/files/media/files/2024-10/citef_brochure_oct_2024_digital_lr.pdf." [Online]. Available: https://www.nexovagroup.eu/sites/default/files/media/files/2024-10/citef_brochure_oct_2024_digital_lr.pdf

[154] D. Conte de Leon, C. E. Goes, M. A. Haney, and A. W. Krings, "ADLES: Specifying, deploying, and sharing hands-on cyber-exercises," *Computers & Security*, vol. 74, pp. 12–40, May 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817302742

[155] 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," Dec. 2024, arXiv:2401.12594 [cs]. [Online]. Available: http://arxiv.org/abs/2401.12594

[156] S. EMK, "ECHO Network." [Online]. Available: https://echonetwork.eu/

[157] B. Nour, M. Pourzandi, R. K. Qureshi, and M. Debbabi, "AUTOMA: Automated Generation of Attack Hypotheses and Their Variants for Threat Hunting Using Knowledge Discovery," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5178–5196, Oct. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10477575

[158] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources," in *Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, December 4-8, 2017*. ACM, 2017, pp. 103–115. [Online]. Available: https://doi.org/10.1145/3134600.3134646

[159] N. Rani, B. Saha, V. Maurya, and S. K. Shukla, "Ttphunter: Automated extraction of actionable intelligence as ttps from narrative threat reports," in *Proceedings of the 2023 Australasian Computer Science Week, ACSW 2023, Melbourne, VIC, Australia, 30 January 2023-3 February 2023*. ACM, 2023, pp. 126–134. [Online]. Available: https://doi.org/10.1145/3579375.3579391

[160] M. T. Alam, D. Bhusal, Y. Park, and N. Rastogi, "Looking beyond iocs: Automatically extracting attack patterns from external CTI," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2023, Hong Kong, China, October 16-18, 2023*. ACM, 2023, pp. 92–108. [Online]. Available: https://doi.org/10.1145/3607199.3607208

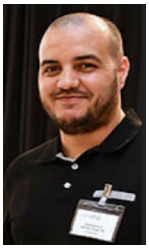
[161] REWIRE Cybersecurity Alliance, "R4.3.1. REWIRE Cyber Range scenario and development framework," Tech. Rep., 2023. [Online]. Available: https://rewireproject.eu/wp-content/uploads/2023/05/R4.3.1-REWIRE_WEB.pdf

[162] "eBPF - Introduction, Tutorials & Community Resources." [Online]. Available: https://ebpf.io

[163] M. Standen, M. Lucas, D. Bowman, T. J. Richer, J. Kim, and D. Marriott, "CybORG: A Gym for the Development of Autonomous Cyber Agents," Aug. 2021, arXiv:2108.09118 [cs]. [Online]. Available: http://arxiv.org/abs/2108.09118

[164] "FORESIGHT-Related Projects." [Online]. Available: https://foresight-h2020.eu/index.php/related-projects/
[165] S. EMK, "18 Months – Milestones of ECHO – ECHO Network." [Online]. Available: https://echonetwork.eu/18-months-milestones-of-echo/
[166] REWIRE Cybersecurity Alliance, "R.5.4.6 Policy Recommendations," Tech. Rep., 2024. [Online]. Available: https://rewireproject.eu/wp-content/uploads/2024/11/R5.4.6_REWIRE_Policy_Recommendations_Website.pdf
[167] G. Sharkov, C. T. Odorova, G. Koykov, and I. Nikolov, "Towards a Robust and Scalable Cyber Range Federation for Sectoral Cyber/Hybrid Exercising: The Red Ranger and ECHO Collaborative Experience," *Information & Security: An International Journal*, vol. 53, pp. 287–302, 2022. [Online]. Available: https://isij.eu/article/towards-robust-and-scalable-cyber-range-federation-sectoral-cyberhybrid-exercising-red
[168] "CYBRTEK, Nortal, and Harrisburg University unite for cybersecurity innovation - Defence & Security Middle East," Oct. 2023, section: Cyber Security. [Online]. Available: https://www.defsecme.com/security/cyber-security/cybrtek-nortal-and-harrisburg-university-unite-for-cybersecurity-innovation

Aayush Garg received his Master's degree in Computer Science (concentration: Security) from Boston University, USA, and completed his PhD in Computer Science (specialization: Quality Assurance and Artificial Intelligence) at the University of Luxembourg. His early career as a Software Engineer in the Fintech sector provided a solid foundation in Software Development and QA. Building on this expertise, he transitioned to a Researcher role at the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg. Currently, Aayush is a Research and Technology Scientist at the Luxembourg Institute of Science and Technology (LIST), Luxembourg. He actively contributes to National and European research initiatives, and his work appears frequently in top international conferences and journals. His primary research interests include Cyber Security, Quality Assurance, and Artificial Intelligence.



Abdelwahab Boualouache is a Senior Research Scientist at the Luxembourg Institute of Science and Technology (LIST) with research interests in the application of Artificial Intelligence (AI) for securing next-generation networks. His work lies at the intersection of AI, cybersecurity, and telecommunications, and he has contributed to numerous collaborative projects involving both academia and industry. His works have been published in top-tier international journals and conferences in these areas.

Adnan Imeri is Senior R&T Scientist at LIST, and he is known for pushing innovation beyond the state of the art in various scientific and industrial domains via applied research in the frame of Research and Development (R&D). He has extensive experience in research activities at the European level, accounting for many international projects, designing, developing, and successfully delivering them. Moreover, industry-related experiences, particularly in the designing, engineering, and architecting of software systems, significantly enrich his career. He, as part of the READY Unit / CYBER research group, is doing applied research related to digital trust, cybersecurity, and privacy in various domains of application. Adnan's recent innovation activities have been primarily centered around DLTs and blockchain, reflecting his adaptability and relevance in the rapidly evolving technological landscape. He holds a PhD in Computer Science from the University of Paris-Saclay (UEVE) and the University of Luxembourg, focusing on blockchain technology and its applicability in real-world use cases. Additionally, he serves as the Technical Lead at Infrachin, further solidifying his standing as a leading figure in the scientific and industrial sectors.

Uwe Roth is a lead research engineer at the Luxembourg Institute of Science and Technology LIST. He is doing research in the CYBER group and is working on all kinds of applied privacy, cybersecurity, and trust-related problems in various domains of application. He contributes to national and European research initiatives and is also a member of the Ethics Committee at LIST. Uwe Roth worked on privacy-related questions in the DLT/Blockchain domain and filed a patent at the end of 2016 that uses a blockchain as a core element to trace the access of files in a distributed hash table network. Uwe Roth completed his PhD in computer science in 2004 and worked as the senior researcher and head of the Interoperability Laboratory for Security in Ad-Hoc Networks of the University of Luxembourg. Starting in 2008 at the Public Research Center Henri Tudor (now LIST), he was the main architect of a national pseudonymisation service that was used by biomedical researchers in Luxembourg to exchange data.