Range42: An Open, Automated, and Extensible Architecture for Next-Generation Cyber Ranges

Abstract

The future of cyber conflict demands training and testing platforms that anticipate tomorrow's threats rather than replicate yesterday's incidents. This paper presents Range42 [1], an open cyber range developed by the National Cybersecurity Competence Center (NC3 [2]) under the Luxembourg House of Cybersecurity [3] together with DIGISQUAD [4]. Designed as a flexible, automated platform, Range42 lowers barriers to building realistic and reproducible training environments while fostering collaborative research and development.

Our research introduces two core contributions and a future standard. First, we describe an extensible catalog system of vulnerable and misconfigured environments. With over 100 curated CVEs and misconfigurations identified and approximately 20 already deployable, this system enables structured reproduction of real-world attack surfaces for controlled experiments. Second, we detail an orchestration framework capable of deploying multi-subnet infrastructures, making it possible to simulate complex enterprise-grade networks with isolation and fidelity. Finally we want to develop a scenario standard language for cyber ranges, well knowing that quite a few efforts already exist yet none englobes the entire process of a cyber range exercise. Bearing this in mind we will develop a baseline on the Common EXercise Format [5], inspect the Open Cyber Range SDL [6] (Scenario Defined Language) and consider the findings of the KYPO Cyber Range [7] We highlight a proof-of-concept LLM-assisted workflow: prompt → scenario narrative + schema → validation → ingestion; this enables fast, potentially innovative and always up to date on the current threat landscape and the most common attacks.

We propose range42-catalog, a modular, open catalog of vulnerable and misconfigured environments, designed for extensibility, interoperability, and community-driven growth. The catalog aligns with our Ansible-based architecture and supports declarative scenario assembly across tools.

We also present a scenario description standard (dual human-readable + machine-ingestible schema) for defining cyber range scenarios (including technical and non-technical injects), enabling a single definition to generate JSON for deployment. We highlight a proof-of-concept LLM-assisted workflow: prompt → scenario narrative + schema → validation → ingestion.

We outline the automation pipeline (Proxmox [8] integration, Ansible [9]-driven orchestration, container/VM deployment, Tailscale-based [10] zero-trust connectivity), and evaluate catalog ingestion, scenario generation, and deployment consistency. We conclude that an open, standardized catalog + scenario architecture is key to "securing tomorrow" through shared scenario ecosystems, faster tool integration, and collaborative innovation.

1 Introduction

Cyber threats evolve faster than training and evaluation methodologies. To secure tomorrow, cyber ranges must enable realistic, reproducible, and rapidly composed environments that reflect emerging attack surfaces and defensive strategies. We present Range42, an open cyber range platform emphasizing automation, flexibility, and collaboration. Our contributions are: (1) an extensible catalog of vulnerable/misconfigured systems for research-grade reproducibility; and (2) an orchestration framework for deploying multi-subnet infrastructures approximating enterprise topologies. We outline our automation pipeline and report lessons for scalability, isolation, and openness. Further a standard scenario description that blends human readability and machine ingestion, enabling consistent narrative and technical deployment definitions is under consideration. We also explore an experimental LLM-assisted pipeline to generate scenarios from prompts; this might be interesting for full automation information gathering. Playing hundreds of generated scenario to extract the generated telemetry might be an interesting source of data. Especially when paired with real-world events and digital twins of certain breaches.

2 Related Work

Existing cyber range catalogs or scenario repositories tend to be closed, bespoke, or tightly coupled to particular platforms. Some efforts provide deployment DSLs or scripts, but few support a human-editable narrative plus structured schema, or offer APIs for third-party tooling. We review virtual scenario description languages, shared scenario repositories, and limitations in openness and interoperability.

3 System Architecture & Catalog Design

Range42 operates via Proxmox, Ansible, and container/VM integration. The range42-catalog is structured as a modular repository of scenario components (e.g. vulnerable hosts, network topologies, inject modules), each described with metadata (e.g. prerequisites, dependencies, scoring hooks). We maintain extension APIs so external tools can list, validate, augment, or contribute entries. The catalog integrates seamlessly with Ansible roles to map component definitions to deployment logic.

4 Vulnerability and Misconfiguration Catalog

We curate an catalog of ∼100 CVEs/misconfigurations (with ∼20 currently deployable), using build descriptors and snapshotting to balance reproducibility with support for proprietary or atypical systems. The design supports traceability, variant creation, and controlled risk exposure during exercises.

5 Scenario Description Standard

Dual-layer standard. We standardize scenarios with a dual representation: (i) a human-readable layer (narrative, learning objectives, context, and both technical and non-technical injects); and (ii) a machine-ingestible schema (JSON) that encodes actors, assets, networks, triggers, dependencies, and scoring. A bidirectional mapping preserves consistency between the narrative and the structured specification.

Lifecycle and validation. Authors draft the human-readable scenario, which is compiled into the machine schema. Automated validators check syntax, dependency closure, resource bounds, and security constraints prior to ingestion. Rejected builds include actionable diagnostics; accepted builds become catalog entries with semantic versioning.

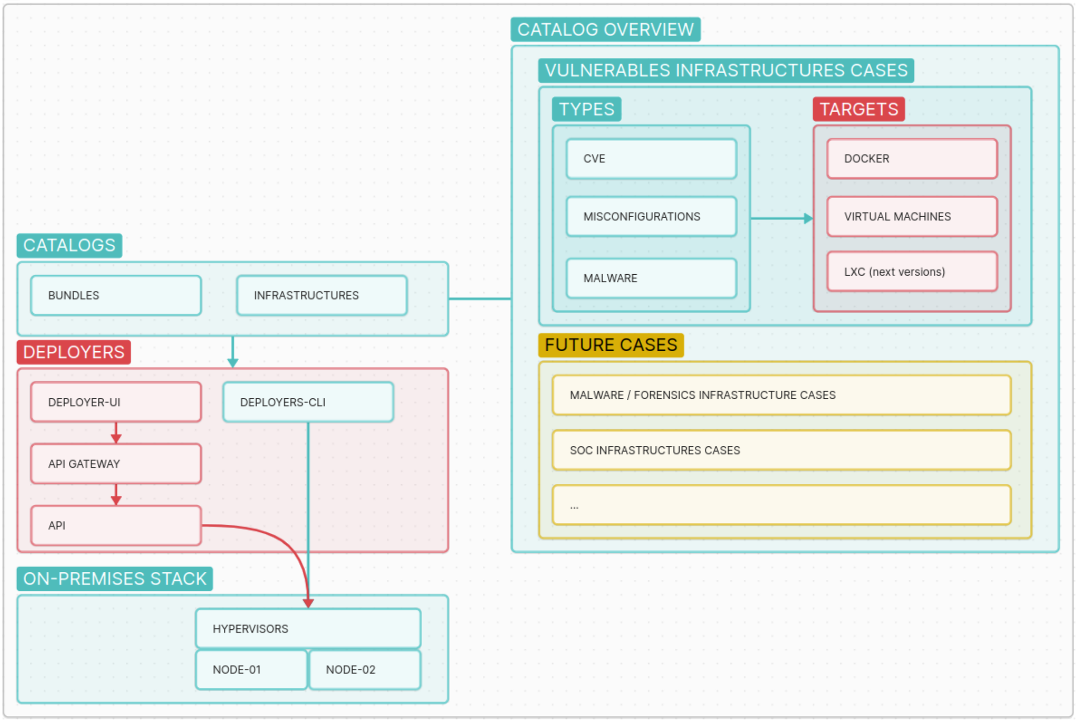

Figure 1: High-level Range42 catalog architecture.

LLM assistance. The standard is designed to leverage LLMs to propose both the human narrative and the machine schema from a prompt (e.g., “simulate a ransomware intrusion across three subnets”). Outputs are never trusted blindly: they pass through the same validation pipeline and require human approval before catalog admission and deployment. We define a domain-specific schema (YAML / JSON hybrid) for scenario narratives. A scenario file includes sections like: background story, objectives, inject schedule, scoring rules, triggers, technical blueprint. From this human-readable form, we generate canonical machine-readable JSON which Range42 ingests to deploy and run the scenario. We include validation logic to catch schema errors or dependency mismatches.

6 Integration & Deployment Pipeline

From catalog + scenario schema to live lab: we describe the orchestration path (Ansible playbooks, Proxmox API calls, network wiring, Tailscale connectivity). We discuss how catalog metadata guides resource allocation, dependency resolution, and runtime coordination of injects and events.

7 LLM-Assisted Scenario Generation

We built a proof-of-concept workflow: given a prompt (e.g. “simulate a ransomware attack in 3-subnet corporate network”), an LLM produces a draft narrative + schema. We feed that into validation logic, correct or flag inconsistencies, then ingest it into the catalog for deployment. Preliminary results and failure modes are discussed.

8 Evaluation

Current deployment status. Range42 has achieved full operational capability in "shooting range" mode, where the complete backend infrastructure is production-ready and actively deployed. The vulnerability catalog currently encompasses approximately 100 identified CVEs and misconfigurations spanning common enterprise technologies, with approximately 20 scenarios available for immediate deployment. The backend automation stack is fully functional, providing end-to-end orchestration from infrastructure provisioning through security monitoring. A graphical user interface for scenario authoring and exercise management is currently under development to complement the operational backend.

The platform features fully automated provisioning on Proxmox infrastructure, including network segmentation, VPN connectivity via Tailscale, and firewall rule deployment. Integrated monitoring through Wazuh provides real-time telemetry collection and alerting capabilities for exercise observation and assessment. The system architecture is distributed across 14 repositories managing Ansible automation playbooks, scenario content, and supporting tooling. In shooting range mode, operators interact with the platform through command-line interfaces and declarative configuration files, enabling rapid scenario deployment for training exercises and research experiments.

Automation maturity. Core automation metrics from internal work packages indicate: hypervisor automation ∼80%, network topology automation ∼70%, and baseline catalog initialization ∼25%. The hypervisor and network automation components represent production-grade capabilities currently in active use, while catalog content development remains an area of ongoing expansion. The term "shooting range" reflects the platform's current operational mode: the backend orchestration, deployment, and monitoring pipelines are fully implemented and battle-tested, whereas user-facing interfaces for non-technical operators are planned enhancements.

Metrics. We report (i) ingestion success rate (schema validation pass/fail), (ii) deployment latency for typical lab topologies (VM- and container-based), and (iii) reproducibility across repeated runs. Where infrastructure constraints limit scale tests, we complement with instructor/operator feedback and failure-mode analyses (e.g., image sprawl, secret handling, routing edge cases). We measure catalog ingestion success rate, scenario consistency (i.e. narrative vs deployed topology), deployment latency, and reusability across runs. Where full metrics are limited, we include qualitative feedback from instructors and early users.

9 Discussion

Governance and federation. A community catalog requires contribution guidelines, schema versioning strategy, and automated QA (linting, policy checks). We envision cross-institution catalog federation where trusted peers exchange signed scenario components, enabling shared curricula and comparable experiments.

Standardization impact. A widely adopted scenario standard lowers authoring friction, enables portable exercises, and allows third-party tools (including LLM-driven assistants) to integrate safely.

Synergies with NGSOTI. We identify strong architectural and conceptual overlap with the Next Generation Security Operator Training Infrastructure (NGSOTI) project [13], which focuses on modular, open architectures for distributed cyber range operations. Range42 aims to cooperate with NGSOTI and re-use compatible components where feasible — particularly around orchestration pipelines, scenario interoperability, and federated data exchange. This alignment ensures that both projects contribute toward a shared European ecosystem of interoperable, open, and extensible cyber training infrastructures.

Table 1: Range42 Implementation Status and Maturity

Component	Status
Catalog & Content	
CVEs & Misconfigurations Cataloged	∼100
Deployable Scenarios	∼20
Backend Infrastructure (Production)	
Hypervisor Automation	80%
Network Topology Automation	70%
Orchestration Repositories	14
Backend Status	Fully Operational
User Interfaces	
CLI/API	Operational
Web GUI	In Development
Integration & Monitoring	
Wazuh Telemetry	Integrated
Tailscale VPN	Integrated
Catalog Initialization	25%

We reflect on expressivity vs. enforceable constraints, schema versioning, contribution governance, catalog federation across platforms, and operational challenges (e.g. schema drift, extension conflicts).

10 Future Work

Catalog. Expand the catalog with richer scenarios (forensics, social engineering, insider threats), build GUI editors for the standard format, integrate stronger LLM-based validation and correction, and establish federated catalogs across multiple range platforms with shared schemas. Advancing a visual lab designer; integrating malware analysis/forensics workflows; and supporting richer hybrid topologies.

Visual lab designer. While Range42's backend orchestration is fully operational in shooting range mode with command-line and API interfaces, the platform requires graphical tools to serve non-technical instructors and students. The visual lab designer will provide drag-and-drop topology composition, allowing users to construct multi-subnet networks, place vulnerable hosts, configure network segmentation, and define inject schedules through an intuitive interface. This component is currently under active development as the Exercise Management Platform (EMP) with mockups completed and frontend implementation in progress.

Exercise management interface. The EMP will extend beyond scenario authoring to provide real-time exercise orchestration, participant monitoring, and automated scoring. Instructors will track student progress, trigger dynamic injects, adjust scenario difficulty, and analyze performance metrics through unified dashboards. Integration with the scenario description standard will enable seamless transitions from design to deployment to assessment.

11 Conclusion

By combining an open catalog architecture with a standardized, dual-mode scenario description and LLM-assisted generation, Range42 is positioned not just as a cyber range but as part of an ecosystem of interoperable scenario tooling. In this context we try to follow the lead of the MISP [12] project when it comes to community building, information sharing, open source ethos and innovation by having open standards. We believe this approach will help catalyze collaborative, scalable, and future-ready cyber range research and training. Automated ranges like Range42 help secure tomorrow by enabling hands-on research and training humans as well as generating data for model generation at scale.

References

[1] The Range#42: https://www.range42.lu/

[2] The National Cybersecurity Competence Center Luxembourg: https://www.nc3.lu/

[3] Luxembourg House of Cybersecurity: https://www.lhc.lu/

[4] DIGISQUAD: https://www.digisquad.com/

[5] MISP Commone EXercise Format: https://github.com/MISP/cexf

[6] Open Cyber Range SDL: https://https://documentation.opencyberrange.ee/docs/sdl/reference/

[7] J. Vykopal, P. Čeleda, P. Seda, V. Švábenský, and D. Tovarňák, "Scalable Learning Environments for Teaching Cybersecurity Hands-on," in 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1–9, doi: 10.1109/FIE44824.2020.9274085.

[8] Proxmox VE. Available: https://www.proxmox.com/

[9] Ansible Documentation. Available: https://docs.ansible.com/

[10] Tailscale Documentation. Available: https://tailscale.com/kb/

[11] Docker Documentation. Available: https://docs.docker.com/

[12] MISP Threat Sharing: https://www.misp-project.org/

[13] Next Generation Security Operator Training Infrastructure: https://d4-project.org/2025/06/19/NGSOTI-Architecture-Overview.html

Use of AI tools and human oversight

The authors disclose that AI-assisted tools were used to aid editing parts of this manuscript, notably for re-organizing content, spellchecking, counter checking examples by contradiction, and checking of stylguide correctness. All substantive technical claims, architectural designs, and empirical data were authored, verified, and approved by the human authors. Automated outputs were reviewed and revised by subject-matter experts to ensure accuracy, safety, and compliance with ethical standards. This acknowledgement is provided in accordance with academic best practices for transparency in the use of AI-assisted tools.