CyCon 2025: The Next Step
17th International Conference on Cyber Conflict
C. Kwan, N. Gratzer, K. Podiņš, M. Tolppa (Eds.)
2025 © NATO CCDCOE Publications, Tallinn

Next Steps in Cyber Blue Team Automation—Leveraging the Power of LLMs

Allard Dijk
Netherlands Defence Academy
Den Helder, The Netherlands
ad.dijk@mindef.nl

Roland Meier
Cyber-Defence Campus
armasuisse
Thun, Switzerland
roland.meier@ar.admin.ch

Cosimo Melella
NATO Cooperative Cyber Defence Centre of Excellence
Tallinn, Estonia
cosimo.melella@ccdcoe.org

Mauno Pihelgas
Tallinn University of Technology
Tallinn, Estonia
maunopihelgas@gmail.com

Risto Vaarandi
Tallinn University of Technology
Tallinn, Estonia
risto.vaarandi@taltech.ee

Vincent Lenders
Cyber-Defence Campus
armasuisse
Thun, Switzerland
vincent.lenders@ar.admin.ch

Abstract: In 2021, driven by the ongoing advancements in artificial intelligence (AI) and automation, previous works [1], [2] introduced architectures for fully automated blue teams in cyber defense exercises such as Locked Shields (LS). Since then, technological and scientific progress has further accelerated. In particular, the rapid evolution of generative AI through large language models (LLMs) has significantly enhanced the capabilities of cybersecurity automation.

This paper reviews how cyber blue team automation can benefit from these recent advances, with a focus on how generative AI and LLMs are reshaping automation strategies for defending complex cyber infrastructure. Using the LS exercise as a case study, we discuss how generative AI-based automation can address the growing complexity of cyber threats. Our paper presents promising directions on how generative AI can enhance fully automated blue teams, and it addresses a major research gap—

the lack of high-quality datasets for training and evaluation in this field. To address this challenge, we introduce a novel dataset containing labeled network traffic and end-host logs, collected during the "partners' run" preceding LS 2024. This dataset is derived from over 400 GB of captured network traffic and more than 6 million log entries. It captures real-world red team behavior and is made publicly available to foster research and AI development in the field of blue team automation.

We conclude with future research challenges in automated cyber defense.

Keywords: *automated cyber defense, Locked Shields, artificial intelligence, large language models, dataset*

1. INTRODUCTION

Artificial intelligence (AI) is disrupting almost every field at an unprecedented pace. This also includes the field of cybersecurity, where AI is transforming both the attack and defense landscapes. While attackers increasingly exploit AI to automate and enhance their cyber exploitation methods, defenders are leveraging AI to improve detection, response, and mitigation strategies. For example, AI can identify suspicious patterns and anomalies in network traffic or application logs, pinpointing potential threats faster and with greater accuracy than traditional methods [3]. Beyond detection, AI is increasingly being used in response automation, such as orchestrating defense mechanisms or remediating vulnerabilities with or without human intervention [4], [5].

However, even though AI has made significant progress, it is not yet advanced enough to fully replace human experts in cyber defense. For instance, AI struggles to adapt to scenarios that deviate from its training data [6]. Furthermore, the potential for false positives and the lack of high-quality labeled data limits the effectiveness of AI in real-world applications. Building on this, Zhang et al. [7] explore the applications of AI in cybersecurity, including user access authentication, network situation awareness, dangerous behavior monitoring, and abnormal traffic identification. They emphasize the role of AI in enhancing cybersecurity measures and propose a conceptual human-in-the-loop cybersecurity model, stressing the importance of human involvement alongside AI systems.

In this paper, we analyze the current capabilities of AI in the context of automating cyber defense. We focus on the use of such automation in live-fire cyber defense

exercises such as Locked Shields (LS), because these exercises provide an ideal testing ground for new technology. Starting with a framework that Meier et al. developed in 2021 [1], we discuss the impact of AI developments that have happened since then and we present the next steps toward the vision of a fully automated defense team at a cyber defense exercise. We base our discussion on ongoing research efforts suitable for LS, the world's largest international live-fire cyber defense exercise. We also publish a labeled dataset from this exercise in order to allow the research community to develop and test their models with realistic data and potentially use it to train or improve LLMs for cybersecurity automation.

In summary, the main contributions of our paper are:

- A retrospective of the latest developments in the context of generative AI and how they affect blue team automation (Section 3);
- A discussion of the main use cases for generative AI for blue team automation (Section 4);
- A plan for the next steps toward the vision of an automated blue team, and the challenges and opportunities that generative AI brings (Section 5);
- A novel dataset containing labeled network traffic and end-host logs to foster research (including training of new LLMs) (Section 6).

2. BACKGROUND ON CYBER DEFENSE EXERCISES

Cyber defense exercises are critical for enhancing operational readiness, fostering interdisciplinary cooperation, and improving cyber defenses in the ever-evolving cyber domain. Among the most prominent examples is LS, an annual live-fire exercise organized by the NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE) since 2010 [8]. It has recently gained additional relevance as a testbed for incorporating AI into cyber defense operations.

LS is a two-day, defense-oriented exercise centered around a fictional geopolitical conflict. Blue teams (BTs), composed of rapid-reaction cybersecurity units, are tasked with defending the IT and critical infrastructure of the fictional country Berylia against the red team (RT), which represents a hostile state, Crimsonia. At a high level, the tasks of the BTs can be grouped into four stages: initial hardening (harden systems before the attacks start), monitoring and response (detect and mitigate attacks), reporting (document the observed attacks), and recovery (restore gamenet systems from backups or with help of the exercise organizers).

The BTs are the main training audience in LS, and they are scored across various categories, including defending against RT attacks, incident reporting, and maintaining service availability. Each BT is responsible for maintaining the uptime and security of over 140 physical and virtual hosts, which include standard IT systems, industrial control systems, and specialized components such as 5G infrastructure.

Recent research [1], [9] has introduced AI into LS, showcasing its potential to enhance defense strategies. Such AI-powered systems can improve defense capabilities by offering faster threat detection and response, scalability to manage complex infrastructures, and continuous learning from attack patterns.

3. THE VISION OF AN AUTOMATED BLUE TEAM

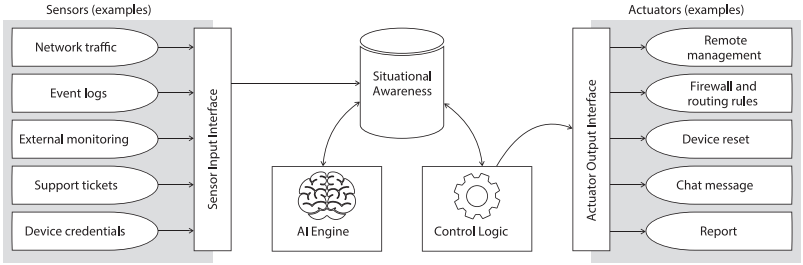
In 2021, Meier et al. developed a general architecture for an AI-powered player in cyber defense exercises [1]. The architecture is depicted in Figure 1. It consists of the following main components:

Sensors are components that provide measurements or data. Examples of sensors and the data that they provide include: network traffic, event logs, device credentials, or support tickets by users.

Actuators are components that perform actions in the gamenet. Examples of actuators include: remote management (e.g., via SSH or RDP), modifications of firewall rules, reset or reboot of a device, or generating a response to a support ticket.

In between the sensors and the actuators are three additional building blocks: the *situational awareness database* (contains all sensor data), the *AI engine* (learns and applies AI models in order to enhance the situational awareness database), and the *control logic* (triggers actuators depending on the contents of the situational awareness database).

FIGURE 1: ARCHITECTURE FOR AN AUTOMATED BLUE TEAM, DEVELOPED BY MEIER ET AL. [1]



In 2021, the authors of [1] could not foresee the upcoming generative AI revolution, most notably marked by the release of ChatGPT in November 2022. This release marked a milestone in the evolution of generative AI, demonstrating its ability to engage in complex, human-like conversations and solve problems.

Fundamentally, generative AI is a type of artificial intelligence designed to create content rather than simply analyze or classify existing data. Generative AI models (e.g., OpenAI's ChatGPT,[1] Google's Gemini,[2] or Meta's Llama[3]) can produce text, images, code, and other creative outputs based on patterns they have learned during training. In the case of LLMs, the focus is on generating coherent, context-aware text that mimics human language.

At their core, LLMs are built on a neural network architecture called Transformers, which excels at processing and generating sequential data, such as language. These models are trained on massive datasets, including books, articles, websites, and other text sources, to identify statistical relationships between words, phrases, and contexts. The goal is not to "understand" language as humans do but to generate text that aligns with patterns and structures found in natural language.

Today's LLMs can produce high-quality outputs and handle a wide range of tasks across industries, including IT and cybersecurity. There, LLMs have proven to be a valuable assistant in areas such as debugging code [10], finding vulnerabilities [11] and analyzing system logs [12]. However, it is important to note that LLMs lack true comprehension or reasoning. Instead, they generate content based solely on learned patterns, and can produce biased or incorrect information if such issues exist in its training data.

4. APPLICATIONS OF LLMS IN BLUE TEAM AUTOMATION

As an AI technology, LLMs primarily influence the "AI engine" component in Figure 1. However, they significantly expand the possibilities for processing sensor data and generating inputs for actuators. In this section, we explore the key use cases where LLMs offer notable advantages over previously available technologies.

We categorize these use cases according to the four stages of a cyber defense exercise: initial hardening, monitoring and response, reporting, and recovery (see Section 2). Table I provides an overview and the remainder of this section explains all use cases in more detail.

1 https://chatgpt.com/
2 https://gemini.google.com/
3 https://llama.com/

TABLE I: OVERVIEW OF USE CASES WHERE LLMS PROVIDE A SIGNIFICANT ADVANTAGE COMPARED TO PREVIOUS METHODS

Stage	Use Cases for LLMs
Initial hardening	• Identification and fixing of vulnerabilities and misconfigurations in software (Section 4.A)
Monitoring and response	• Analyzing network traffic for malicious activities (Section 4.B) • Analyzing event logs for malicious activities (Section 4.C) • Parse support tickets, trigger corresponding actions, and generate responses (Section 4.D) • Generate commands and configurations for remote management (Section 4.E)
Reporting	• Link incidents to IoCs and generate human-readable reports (Section 4.F) • Generate human-readable reports required for the exercise (e.g., post-incident summaries) (Section 4.E)
Recovery	• Identifying and documenting affected systems for recovery prioritization (Section 4.C) • Reverting devices, misconfigurations, and patch failures using rollback mechanisms such as backups and snapshots (Section 4.E) • Generating comprehensive post-recovery analysis and lessons learned documentation (Section 4.F)

A. Detecting vulnerabilities and misconfigurations

Software vulnerabilities are flaws or weaknesses in an application's design, implementation, or configuration. In an exercise like LS, these weaknesses can include anything from poorly secured web applications and misconfigured Docker containers to hidden backdoors intentionally placed by the RT. For a BT, discovering and remediating these vulnerabilities quickly is vital.

Traditional methods have proven effective but are often time-consuming and demand specialized expertise. For example, traditional static analysis techniques (see [13]) have uncovered numerous bugs at scale, but they struggle to keep pace with increasingly complex systems. Similarly, dynamic taint analysis [14] set early precedents for automated exploit detection but faces scalability issues in modern environments.

LLM-based approaches offer greater flexibility and efficiency. Systems like LProtector, built on GPT-based models, excel at detecting vulnerabilities in large codebases [15]. By training on extensive code repositories, these models can identify issues like SQL injection, remote code execution, and cross-site scripting with remarkable accuracy [16], [17]. At the same time, the use of AI-driven code generation tools (e.g., GitHub Copilot) has been scrutinized for potential security risks [18].

LLMs can similarly detect misconfigurations by examining database or web server settings, pinpointing insecure network parameters or permissive access controls [19]. This proactive approach helps preempt exploitation by simulating possible attack vectors.

Research also suggests that AI-driven rule adjustments for security policies can keep pace with emerging threats [20]. By prioritizing vulnerabilities according to severity, defenders can allocate resources more effectively [15]. Finally, while direct LLM-based remediation remains an emerging topic, previous efforts in machine-learning-driven security automation indicate a promising direction [21].

B. Network traffic analysis

With the integration of data-mining-based algorithms and, more recently, LLMs, the field of network traffic analysis has seen significant advancements. Traditional algorithms such as decision trees or support vector machines were employed to analyze traffic for detecting patterns and anomalies [22], [23].

LLM-based approaches introduced a new paradigm in traffic analysis: LLMs can process and understand vast amounts of unstructured data (e.g., network logs) and automate incident response actions [24]. They can recommend or autonomously execute predefined responses to threats, reducing the time and effort required from human analysts.

LLM-based methods can also classify different malware types with limited amounts of training data compared to state-of-the-art methods: Even though the structure of network protocols is different from natural language, Stein et al. [25] demonstrate that transformer-based models can capture and learn the intricate sequential patterns. Unlike many LLMs that are pre-trained and then fine-tuned, RTIDS [26] shows that a transformer-based intrusion detection system (IDS) can achieve promising results when trained from scratch by batching collections of network flows during the training process. However, such supervised approaches require extensive labeled datasets, which can be challenging to obtain. The LSPR23 [9] dataset and the LSPR24 dataset released with this paper facilitate this kind of research.

C. Log analysis

In the last two decades, several data-mining-based algorithms have been proposed for analyzing textual event logs (cf. [27]). These algorithms focus on the detection of line patterns or templates from textual event logs, with the following example illustrating two example log messages and a template representing them:

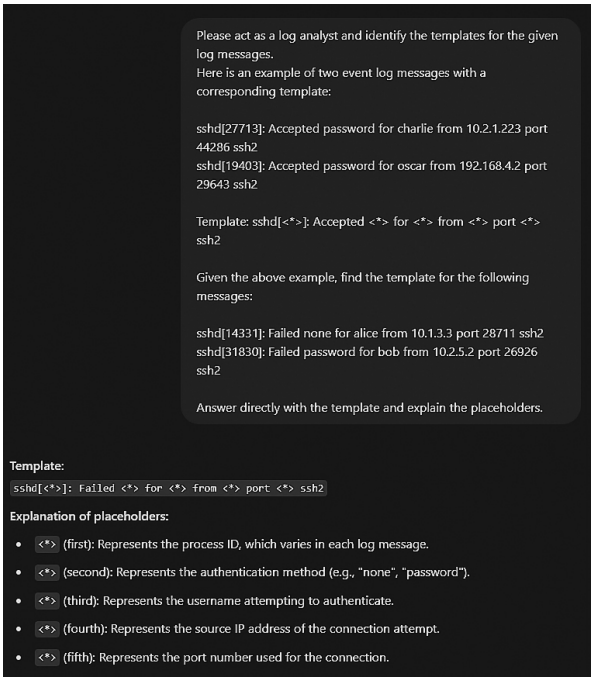
```
sshd[27713]: Accepted password for charlie from 10.2.1.223 port 44286 ssh2
sshd[19403]: Accepted password for oscar from 192.168.4.2 port 29643 ssh2
Template: sshd[<*>]: Accepted <*> for <*> from <*> port <*> ssh2
```

The purpose of the template detection is to identify variable parts in event logs and replace these parts with a wildcard (<*> in the above example). This is useful for detecting event types from logs and assisting the development of event parsing rules. Additionally, these insights can support the identification and documentation of affected systems in a cyber exercise, enabling prioritization in recovery efforts by understanding the scope and impact of incidents.

Recent studies have demonstrated the potential of using LLMs for template detection tasks [28], [29], [30], [31], [32], [33]. Although LLM-based approaches require more computational resources than traditional data-mining-based algorithms and so tend to be slower [28], [32], [33], they have several benefits. For example, some LLMs have the ability to infer a correct template even from insufficient log data [33].

Some algorithms like LLMParser [29] and LogPPT [31] use *fine-tuning* of local LLMs, which involves additional training of LLMs with examples of event log messages and expected templates. The other and more commonly used approach is *in-context learning*, which involves providing an LLM with instructions (*prompt*) on the template detection task [28], [30], [32], [33]. Usually, the prompt contains some examples of event log messages with expected template(s) and the actual event log messages. Figure 2 shows an example using ChatGPT.

FIGURE 2: LLM PROMPT TO EXTRACT A TEMPLATE FROM LOG MESSAGES (USING CHATGPT 4O)



Since the response from an LLM is provided in natural language, the algorithms that are using LLMs through in-context learning must parse the LLM's answer in order to identify the templates in the received response.

Algorithms that rely on in-context learning can be supervised or unsupervised. Existing supervised algorithms LILAC [28] and DivLog [30] assume that the human expert has to create a larger set of example log messages with a correct template provided for each message. When constructing the prompt, the algorithms analyze the event log messages supplied by the user, and select the most appropriate examples from the set prepared by the human expert. The main drawback of supervised algorithms is the need for such data sets with expert-supplied templates. LUNAR [32] and LLM-TD [33] are unsupervised algorithms which do not employ large, manually created example sets to build prompts but rather use prompts with static instructions and examples. LLM-TD mines *syslog* messages, whereas LUNAR employs a hierarchical clustering algorithm to detect similar messages that are suitable for submitting to the LLM in one query.

From the aforementioned algorithms that rely on in-context learning, DivLog, LILAC, and LUNAR employ public LLMs (e.g., ChatGPT through the OpenAI interface). Since LLM-TD has been specifically designed for analyzing security event logs, it uses local LLMs through the Ollama framework in order to avoid submitting potentially sensitive log data to external service providers.

D. Interacting with humans

In a cyber defense exercise, BTs typically handle a continuous influx of user inquiries, status updates, and incident reports. Traditionally, these tasks were allocated to human analysts who had to parse support tickets and either execute relevant technical actions or delegate issues to other specialized team members.

LLMs bring efficiency to this process by interpreting the natural-language content of support tickets, extracting critical information (e.g., IP addresses, error codes, account names), and aligning them with the corresponding technical actions [34]. For instance, an LLM-based system may scan a high volume of tickets, identify distinct categories such as “hardware failures” or “phishing suspicions,” and automatically open an internal task for resetting credentials or blocking malicious domains [35]. Thereby, LLMs significantly compress the review cycle time [36].

LLMs can also create human-readable summaries and incident reports. Rather than manually drafting a lengthy post-incident description, analysts can rely on the LLM to compile system logs, relevant indicators of compromise (IoCs), and incident timelines into a coherent narrative [37]. In exercises that score teams on thorough and timely incident reporting, this functionality ensures both clarity and consistency, thereby reducing the risk of miscommunication [38].

E. Remote management

Beyond assisting human interactions, LLMs also play a pivotal role in controlling the infrastructure directly. IT environments require configuration files, scripts, or remediation commands to be maintained in real time [39]. Handling this efficiently can be challenging, especially under the time pressure of a live-fire exercise where multiple systems need simultaneous updates or patches [40].

LLMs can convert high-level policy descriptions or abstract instructions into code or commands, enabling the automated generation of restoration scripts and configurations needed to recover compromised or misconfigured systems [41], [42]. For instance, when the control logic component flags an unauthorized process on a critical server, the LLM can propose a suitable script to terminate that process, quarantine files, or modify a firewall configuration [43]. This eliminates the need for a human operator to research appropriate syntax or recall rarely used commands. In addition, by integrating

with version control repositories, an LLM can track system configurations over time, offering automated rollbacks if an action inadvertently disrupts legitimate services [42].

A particularly promising avenue involves coupling LLMs with "computer use" modes, where the LLM can directly interface with network devices or cloud-based management consoles. In this scenario, the language model constructs the commands, verifies them against known best practices or policy constraints, and then executes them autonomously or with minimal supervision [44]. While this streamlines remote management, it also raises questions about access controls and the risk of an attacker manipulating the LLM to issue malicious commands [45].

F. Integrating threat intelligence feeds and SIEM systems

Leveraging external threat intelligence feeds, such as those provided by the Malware Information Sharing Platform (MISP) [46], is critical for enhancing cybersecurity workflows by enabling the sharing of IoCs and fostering collaboration [47]. LLMs offer a transformative approach to processing and integrating this data into security information and event management (SIEM) systems by automating tasks like ingestion, contextualization, and prioritization [48].

In scenarios like LS, LLMs could dynamically analyze threat intelligence feeds, categorize threats by severity, and link related IoCs to broader campaigns, providing actionable insights that improve situational awareness and decision-making [49]. Integrating external intelligence feeds with SIEMs through LLMs creates a pipeline for correlating IoCs with internal logs, ranking threats by relevance, augmenting data with contextual analysis, and suggesting automated responses, such as blocking IPs or quarantining devices [50].

This synergy reduces the load on analysts, enhances detection speed, and facilitates post-event analysis by generating comprehensive reports. Furthermore, it supports the creation of detailed post-recovery analysis and lessons learned documentation, ensuring organizations can refine their defensive measures based on past incidents [51]. Despite these advantages, challenges include ensuring data quality, maintaining privacy through locally hosted or fine-tuned models, and addressing interpretability issues in LLM outputs to justify their decisions [52]. Experimental frameworks combining MISP, SIEMs, and LLMs could provide valuable insights into real-world applications, paving the way for more efficient and automated cyber defense [53].

5. CHALLENGES AND NEXT STEPS

Based on the insights from the previous sections, we now discuss the current challenges at a higher level and outline the next steps toward the vision of an automated BT.

A. Challenges

Data availability: Training or fine-tuning LLMs requires extensive and high-quality datasets. Without sufficient data, models may underperform or fail to generalize effectively. Cyber defense exercises such as LS provide a good basis for gathering high-quality training data. However, it is also important to be aware of the differences between exercises and real-world incidents where attacks are more subtle and leverage a larger set of strategies.

Prompt engineering: Well-crafted prompts are critical for guiding LLM behavior. This is especially a challenge because the ultimate vision is for these prompts to be generated automatically, without human involvement. A related challenge is the so-called context size of an LLM. This refers to the maximum amount of information it can process at once (i.e., its capacity to "remember"). If an LLM needs to process vast amounts of information (e.g., log files or network data), preprocessing is required to only provide the LLM the relevant information.

Hallucination: Hallucination of LLMs refers to their tendency to generate inaccurate or fabricated information and is difficult to identify. This occurs in any LLM application, but in the context of a fully automated system, it can have greater consequences because there is no "human in the loop" who could detect the hallucination.

Integration complexity: Implementing seamless interfaces between the various components (see Figure 1) is a significant engineering challenge.

Computational power: Running LLMs can demand substantial computational resources. However, there are promising alternatives, such as models optimized for commercial off-the-shelf GPUs, and cloud-based models that can mitigate this issue.

Measuring effectiveness: Assessing the performance of an automated BT (and its components, including the LLMs) in a reproducible manner is critical. While cyber defense exercises like LS provide a valuable opportunity for such experiments, the fact that these exercises typically happen only once a year slows progress. Ideally, there should be a reproducible environment for testing systems multiple times per year.

B. Next steps

To integrate LLMs into an automated BT, we propose the following next steps:

Integrate and evaluate individual LLM components: We estimate that the following use cases have the highest potential for LLMs to achieve a significant advantage compared to traditional solutions. Therefore, they should be addressed first:

- Automating support ticket processing: Utilize LLMs to convert human-written support tickets into actionable technical instructions, such as code, commands, or configuration files.
- Generating human-readable reports: Leverage LLMs to create detailed, easily understandable reports or responses to support tickets.
- Detecting and fixing misconfigurations: Use LLMs to identify system misconfigurations and generate precise corrective actions, including code or commands.
- Combining data for actionable insights: Employ LLMs to analyze and synthesize data from multiple sources, such as event logs and network traffic, to uncover valuable insights and patterns.

Establish a reproducible testing environment: To allow for consistent evaluation and improvement of the automated BT, there needs to be a testing environment that supports frequent, repeatable tests. To maximize efficiency, the testing environment should operate without requiring manual actions from human experts (e.g., from an RT), as such resources are often difficult to obtain. Instead, the environment could leverage automated scenarios, potentially running within a cyber range to simulate realistic attack-defense-interactions. Furthermore, RT automation is a related field of research that we did not cover in this paper. However, such a testing environment could serve as a playground for experimenting with automated RTs versus automated BTs, fostering advancements in both areas and enabling comprehensive evaluations of emerging defensive and offensive strategies.

6. THE LSPR24 DATASET

Collected during the "Partners' Run" prior to Locked Shields 2024, the LSPR24 dataset provides a solid foundation for BT automation research. We publish it to facilitate AI-driven model training by the research community.[4] The dataset also enables validation of automated frameworks that integrate logs from multiple sources, and its structure allows for more effective log analysis and automated responses, particularly in combination with LLMs.

4 https://doi.org/10.5281/zenodo.14900873

A key feature of LSPR24 is that it originates from a complex, realistic environment. Spanning over 400 GB of captured network traffic, it represents diverse hardware configurations, software stacks, and user behaviors, making it a robust resource for machine learning. Host logs, network flows, and Suricata/Zeek outputs help researchers observe both benign and malicious behaviors, including lateral movement and command-and-control (C2) methods.

Figure 3 presents the LSPR24 high-level network map for LSPR24, connecting the government, military, and energy sectors. It integrates advanced technologies like 5G, AI surveillance, and hybrid-cloud systems with traditional satellite communication, air defense, and border security.

FIGURE 3: HIGH-LEVEL OVERVIEW OF THE GAMENET IN WHICH WE CAPTURED THE LSPR24 DATASET

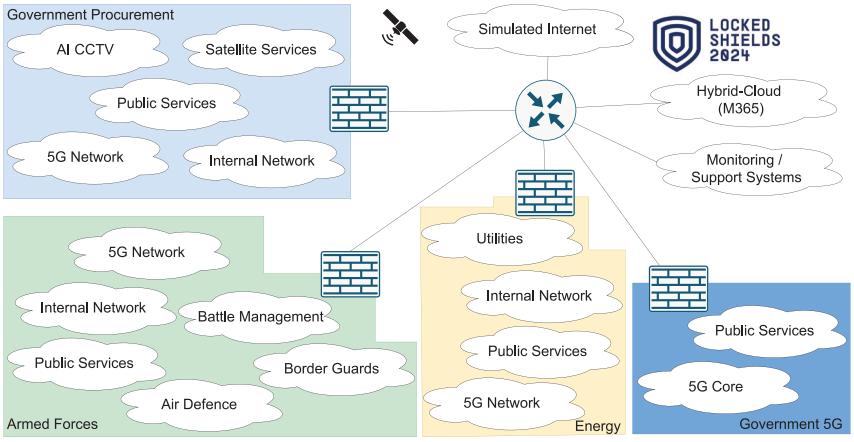


FIGURE 4: HOURLY ACTIVITIES IN THE LSPR24 DATASET

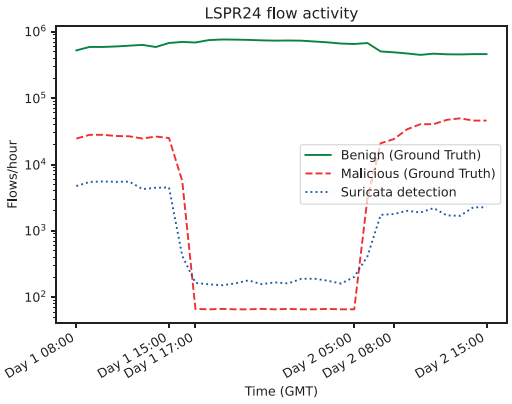


Figure 4 shows the flow activity over the course of the dataset. Benign traffic (green) remains consistently high—on the order of 1 million flows per hour—while malicious traffic (red) shows more fluctuation. Notably, it dips sharply around Day 1 17:00 GMT, then intensifies again on Day 2. The detections of Suricata, a popular traditional intrusion-detection system, (blue) show many false positives (during times when there is no malicious activity). This serves as an indication that more sophisticated technology is needed to detect attacks.

LSPR24 contains 20 million flows, two billion packets, and 287 GB of transferred data over 31.6 hours. The collection spans activity across 13,000 IPv4 and IPv6 addresses, including 372 linked to the RT.

Compared to its predecessor, LSPR23 [9], LSPR24 addresses gaps in IDS signatures, including those targeting Cobalt Strike beacon traffic. It also improves internal flow labeling to accurately classify stepping-stone attacks, enhancing the analysis of suspicious behavior within a defended network.

7. CONCLUSION

This paper revisited the vision of a fully automated blue team (BT), originally published in 2021, and explored how advancements in generative AI can contribute to realizing this vision. By examining the potential applications of generative AI in cyber defense, we identified both opportunities and challenges that remain in the field. A key practical obstacle is the availability of high-quality datasets necessary for the development and evaluation of AI models. To address this gap and foster further research, we published a new labeled dataset comprising network flows and event logs collected during Locked Shields, the world's largest live-fire cyber defense exercise.

By providing insights into generative AI's role and offering resources to the research community, this paper serves as a foundation and guideline for advancing toward the goal of a fully automated BT. Future research should focus on addressing the highlighted challenges and building on the resources provided to achieve this vision.

Notably, generative AI is not only useful for BTs but also for Red Teams (RTs), potentially creating a new dynamic between increasingly automated adversarial and defensive systems [54]. While this paper focused on the BT perspective, the interplay between BT and RT automation creates another relevant research direction for the future.

ACKNOWLEDGMENTS

Risto Vaarandi's work was supported by the Estonian Centre of Excellence in Artificial Intelligence (EXAI), funded by the Estonian Ministry of Education and Research grant TK213.

REFERENCES

[1] R. Meier, A. Lavrenovs, K. Heinäaro, L. Gambazzi, and V. Lenders, "Towards an AI-powered player in cyber defence exercises," in *Proc. 13th Int. Conf. Cyber Conflict (CyCon)*, Tallinn, Estonia, May 2021.

[2] A. Kott, Ed., *Autonomous Intelligent Cyber Defense Agent (AICA): A Comprehensive Guide, , Advances in Information Security*, vol. 87. Cham: Springer, 2023.

[3] I. H. Sarker, M. H. Furhad, and R. Nowrozy, "AI-driven cybersecurity: An overview, security intelligence modeling and research directions," *SN Comput. Sci.*, vol. 2, Mar. 2021.

[4] S. Lysenko, "The role of artificial intelligence in cybersecurity: Automation of protection and detection of threats," *Econ. Aff.*, vol. 69, Feb. 2024.

[5] L. Alevizos, "Automated cybersecurity compliance and threat response using AI, blockchain and smart contracts," *Int. J. Inf. Technol.*, Dec. 2024.

[6] L. Gehri, R. Meier, D. Hulliger, and V. Lenders, "Towards generalizing machine learning models to detect command and control attack traffic," in *Proc. 15th Int. Conf. Cyber Conflict: Meeting Reality (CyCon)*, Tallinn, Estonia, May 2023.

[7] Z. Zhang et al., "Artificial intelligence in cyber security: Research advances, challenges, and opportunities," *Artif. Intell. Rev.*, vol. 55, Feb. 2022.

[8] NATO Cooperative Cyber Defence Centre of Excellence, "NATO Locked Shields." Accessed: Jan. 4, 2025. [Online]. Available: https://ccdcoe.org/exercises/locked-shields/

[9] A. Dijk, E. Halisdemir, C. Melella, A. Schu, M. Pihelgas, and R. Meier, "LSPR23: A novel IDS dataset from the largest live-fire cybersecurity exercise," *J. Inf. Secur. Appl.*, vol. 85, Sep. 2024.

[10] S. S. Sengar, A. B. Hasan, S. Kumar, and F. Carroll, "Generative artificial intelligence: A systematic review and applications," *Multimed. Tools Appl.*, Aug. 2024.

[11] Z. B. Akhtar, "Unveiling the evolution of generative AI (GAI): A comprehensive and investigative analysis toward LLM models (2021–2024) and beyond," *J. Electr. Syst. Inf. Technol.*, vol. 11, Jun. 2024.

[12] E. Karlsen, X. Luo, N. Zincir-Heywood, and M. Heywood, "Benchmarking large language models for log analysis, security, and interpretation," *J. Netw. Syst. Manag.*, vol. 32, Jul. 2024.

[13] A. Bessey et al., "A few billion lines of code later: Using static analysis to find bugs in the real world," *Commun. ACM*, vol. 53, Feb. 2010.

[14] J. Newsome and D. X. Song, "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software," in Proc. *Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2005.

[15] Z. Sheng, F. Wu, X. Zuo, C. Li, Y. Qiao, and L. Hang, "LProtector: An LLM-driven vulnerability detection system," Nov. 14, 2024, *arXiv:2411.06493*.

[16] M. Siavvas, I. Kalouptsoglou, E. Gelenbe, D. Kehagias, and D. Tzovaras, "Transforming the field of vulnerability prediction: Are large language models the key?," in *Proc. 32nd Int. Conf. Modeling, Anal. Simul. Comput. Telecommun. Syst. (MASCOTS)*, Krakow, Poland, 2024, pp. 1–6, doi: 10.1109/MASCOTS64422.2024.10786575.

[17] J. Haurogné, N. Basheer, and S. Islam, "Vulnerability detection using BERT based LLM model with transparency obligation practice towards trustworthy AI," *Mach. Learn. Appl.*, vol. 18, Dec. 2024.

[18] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions," presented at the 2022 *IEEE Symp. Secur. Priv. (SP)*, May 2022.

[19] C. Asuai and G. Nwalozie, "Investigating and addressing security policy misconfigurations," *IOSR J. Eng.*, vol. 14, Apr. 2024.

[20] C. Benzaïd and T. Taleb, "AI for beyond 5G networks: A cyber-security defense or offense enabler?," *IEEE Netw.*, vol. 34, Nov. 2020.

[21] U. Mandal, S. Shukla, A. Rastogi, S. Bhattacharya, and D. Mukhopadhyay, "µLAM: A LLM-powered assistant for real-time micro-architectural attack detection and mitigation," *Cryptol. ePrint Arch*., Paper 2024/1978, 2024. [Online]. Available: https://eprint.iacr.org/2024/1978
[22] A. Dijk, "Detection of advanced persistent threats using artificial intelligence for deep packet inspection," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2021.
[23] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, and N. Tihanyi, "Generative AI and large language models for cyber security: All insights you need," May 21, 2024, *arXiv:2405.12750*.
[24] C.-N. Hang, P.-D. Yu, R. Morabito, and C.-W. Tan, "Large language models meet next-generation networking technologies: A review," *Future Internet*, vol. 16, Oct. 2024.
[25] K. Stein, A. A. Mahyari, G. F. III, and E. El-Sheikh, "Towards novel malicious packet recognition: A few-shot learning approach," Sep. 17, 2024, *arXiv:2409.11254*.
[26] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "RTIDS: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, 2022.
[27] Z. A. Khan, D. Shin, D. Bianculli, and L. Briand, "Guidelines for assessing the accuracy of log message template identification techniques," in *Proc. 44th Int. Conf. Softw. Eng. (ICSE)*, Jul. 2022.
[28] Z. Jiang et al., "LILAC: Log parsing using LLMs with adaptive parsing cache," *Proc. ACM Softw. Eng.*, vol. 1, Jul. 2024.
[29] Z. Ma, A. R. Chen, D. J. Kim, T.-H. P. Chen, and S. Wang, "LLMParser: An exploratory study on using large language models for log parsing," presented at the 2024 *IEEE/ACM 46th Int. Conf. Softw. Eng. (ICSE)*, Apr. 2024.
[30] J. Xu, R. Yang, Y. Huo, C. Zhang, and P. He, "DivLog: Log parsing with prompt enhanced in-context learning," in *Proc. IEEE/ACM 46th Int. Conf. Softw. Eng. (ICSE)*, Apr. 2024.
[31] V.-H. Le and H. Zhang, "Log parsing with prompt-based few-shot learning," in *Proc. 45th Int. Conf. Softw. Eng. (ICSE)*, Jul. 2023.
[32] J. Huang, Z. Jiang, Z. Chen, and M. R. Lyu, "LUNAR: Unsupervised LLM-based log parsing," Aug. 2024, *arXiv:2406.07174*.
[33] R. Vaarandi and H. Bahsi, "Using large language models for template detection from security event logs," *Int. J. Inf. Security*, vol. 24, Mar. 2025.
[34] W. Zhou et al., "Star: A system for ticket analysis and resolution," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min*., 2017.
[35] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, "SecureBERT: A domain-specific language model for cybersecurity," Oct. 20, 2022, *arXiv:2204.02685*.
[36] N. Arici, L. Putelli, L. Sigalini, I. Serina, and others, "LLM-based approaches for automatic ticket assignment: A real-world Italian application," in *CEUR Workshop Proc*., vol. 3551, Nov. 2023.
[37] F. Y. Loumachi and M. C. Ghanem, "Advancing cyber incident timeline analysis through rule-based AI and large language models," Sep. 2024, *arXiv:2409.02572*.
[38] P. Balasubramanian, J. Seby, and P. Kostakos, "CYGENT: A cybersecurity conversational agent with log summarization powered by GPT-3," Mar. 25, 2024, *arXiv:2403.17160*.
[39] F. Li, H. Lang, J. Zhang, J. Shen, and X. Wang, "PreConfig: A pretrained model for automating network configuration," Mar. 14, 2024, *arXiv:2403.09369*.
[40] O. G. Lira, O. M. Caicedo, and N. L. S. da Fonseca, "Large language models for zero touch network configuration management," Aug. 23, 2024, *arXiv:2408.13298*.
[41] Y. Mikami, A. Melnik, J. Miura, and V. Hautamäki, "Natural language as policies: Reasoning for coordinate-level embodied control with LLMs," Apr. 6, 2024, *arXiv:2403.13801*.
[42] K. Dzeparoska, J. Lin, A. Tizghadam, and A. Leon-Garcia, "LLM-based policy generation for intent-based management of applications," in *Proc. 19th Int. Conf. Netw. Serv. Manag. (CNSM)*, Oct. 2023.
[43] S. Hays and J. White, "Employing LLMs for incident response planning and review," Mar. 2, 2024, *arXiv:2403.01271*.
[44] R. Kaur, T. Klobučar, and D. Gabrijelčič, "Harnessing the power of language models in cybersecurity: A comprehensive review," *Int. J. Inf. Manag. Data Insights*, vol. 5, Jun. 2025.
[45] OWASP, "OWASP Top 10 for LLM Applications 2025," *OWASP Top 10 for LLM & Generative AI Security*. Accessed: Jan. 4, 2025. [Online]. Available: https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
[46] MISP Project, "MISP." Accessed: Jan. 4, 2025. [Online]. Available: https://www.misp-project.org/
[47] P. Rafiey and A. Namadchian, "Using LLMs as AI agents to identify false positive alerts in security operation center," *Research Square*, Nov. 2024, doi: 10.21203/rs.3.rs-5420741/v1.
[48] M. Kaheh, D. K. Kholgh, and P. Kostakos, "Cyber Sentinel: Exploring conversational agents in streamlining security tasks with GPT-4," Sep. 28, 2023, *arXiv:2309.16422*.

[49] T. Ali and P. Kostakos, "HuntGPT: Integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs)," Sep. 27, 2023, *arXiv:2309.16021*.
[50] O. Oniagbi, "Evaluation of LLM agents for the SOC Tier 1 analyst triage process," M.S. thesis, University of Turku, 2024.
[51] A. Baig, "Accessing the role of artificial intelligence in information security risk management," M.S. thesis, University of Jyväskylä, 2024.
[52] L. Wang et al., "From sands to mansions: Enabling automatic full-life-cycle cyberattack construction with LLM," Jul. 24, 2024, *arXiv:2407.16928*.
[53] E. Pleshakova, A. Osipov, S. Gataullin, T. Gataullin, and A. Vasilakos, "Next gen cybersecurity paradigm towards artificial general intelligence: Russian market challenges and future global technological trends," *J. Comput. Virol. Hacking Tech.*, vol. 20, Sep. 2024.
[54] M. Cadrouil, "LLM agents in cybersecurity: A double-edged sword," I-TRACING. Accessed: Jan. 4, 2025. [Online]. Available: https://i-tracing.com/blog/llm-agents-cybersecurity/