A Lightweight Emulation Infrastructure for Behavioral Threat Modeling using MITRE ATT&CK*

Amos AKOGBE[1,*,†], Emery ASSOGBA[2]

[1]Carnegie Mellon University Africa, Kigali, Regional ICT Center of Excellence Building
[2]Institut de Mathématiques et de Sciences Physiques (IMSP), University of Abomey-Calavi, Benin

Abstract

Recent threat intelligence reports highlight a significant increase in sophisticated cyberattacks targeting B2B institutions worldwide. To counter these evolving threats, security operations are shifting from traditional indicator-based defenses toward behavior-based detection, utilizing tactics, Techniques, and Procedures (TTPs) formalized by the miter ATT&CK framework. However, training and modeling these behaviors require high-fidelity emulation environments, which are often resource-heavy and inaccessible to researchers in emerging tech hubs. In this paper, we propose Vuln_Box, a lightweight container-based Infrastructure-as-Code (IaC) laboratory powered by the kathará emulation engine. We demonstrate the efficacy of this infrastructure by simulating the complete life cycle of a modern cross-domain exploit (Spring4Shell) and mapping it to the MITRE ATT&CK matrix. Our results provide a granular behavioral threat model for defensive gap analysis and show that the containerized approach drastically reduces resource overhead (RAM, CPU, and boot time) compared to traditional hypervisors and kubernetes clusters, democratizing access to advanced cybersecurity research.

Keywords

Kathará, MITRE ATT&CK, Threat modeling,

1. Introduction

The global threat landscape has reached an unprecedented scale, and Microsoft reported more than 600 million targeted attacks in 2024 alone [1]. This surge necessitates a shift from reactive security to proactive penetration testing. However, there is a significant barrier in emerging tech hubs, such as Benin, where access to realistic, high-fidelity training environments is limited. This accessibility gap prevents cybersecurity professionals from developing the hands-on skills required to counter modern sophisticated exploits. Another aspect of this issue is the critical targets observed by many of these attacks. In fact, several researches revealed that some of the major cybersecurity data breaches and attacks that have recently occurred have been targeting and exploiting smart technologies such as AI, IoT systems and personal identifiable information [2]. It is evident that data and these new technologies are directly connected to the existence and growth of smart cities.[3] Therefore, the security of smart cities needs to be even more protected and is even more critical.

Although frameworks like MITRE's provide a robust taxonomy for adversarial behavior, there is a lack of lightweight, reproducible, and vendor-neutral infrastructures that allow researchers to simulate the entire lifecycle of modern vulnerabilities. Specifically, current educational labs often fail to illustrate the transition from a web exploit (like Spring4Shell) to post-exploitation phases. Therefore, this research addresses this challenge by designing an Infrastructure-as-Code (IaC) laboratory using kathará to analyze cross-domain attack chains (web, network, and host) through the lens of the MITRE ATT&CK framework with a final threat model for these attacks.

The methodology employed in this research follows a multi-phased approach, beginning with a comprehensive literature review during which we analyze the cybersecurity state of art over the last decade. This is followed by an experimental phase that involves the design and deployment of a virtualized

Cotonou '26: CITA 2026 - Land, Smart Cities, and Sustainable Development, 25-26 June 2026, Cotonou, Benin
*Corresponding author.
†These authors contributed equally.
aakogbe@andrew.cmu.edu (A. AKOGBE); emery.assogba@imsp-uac.org (E. ASSOGBA)
https://w4lk3r04.github.io/ (A. AKOGBE); https://dblp.org/pid/277/6019 (E. ASSOGBA)

infrastructure using the kathará emulation engine to validate adversarial behaviors against the MITRE ATT&CK framework.

The main contributions of this paper are threefold:

1. The design of a lightweight, reproducible 3-tier network topology using kathará containerization.
2. A systematic threat modeling methodology bridging raw Spring4Shell exploits to standardized MITRE ATT&CK TTPs.
3. Performance evaluation demonstrating the computational efficiency of the emulation environment compared to traditional virtual infrastructures.

2. Background and related work

2.1. Behavioral threat modeling frameworks

Anticipating and accurately attributing cyberattacks before organizational impact remains a primary objective for Security Operations Centers (SOCs). The "Pyramid of Pain" concept illustrates the paradigm shift in modern security: relying on easily modifiable indicators (like IP addresses or hashes) is insufficient. Consequently, robust defense strategies now prioritize Behavior-based detection focusing on an adversary's Tactics, Techniques, and Procedures (TTPs).[4] MITRE ATT&CK, the Cyber Kill Chain and the Diamond Model of analysis are the most common and effective frameworks that are used for threat and attack analysis. Based on its rich and structured model, MITRE ATT&CK offers an efficient and granular overview and model of any threat. In addition, we have found that its regular updates also guaranty a maximum attack coverage, including for the most recent attacks [8].

Table 1
Comparison of the main frameworks for analyzing computer attacks

Analysis Criteria	MITRE ATT&CK	Cyber Kill Chain	Diamond Model
Documentation of TTP	Comprehensive documentation	General documentation	Limited documentation
Scope	Extensive	Not very extensive	Not very extensive
Ease of use	Very simple	Slightly complex	Slightly complex
Interoperability	Possible	Possible	Possible
Adoption	Very popular	Popular	Not very popular
Updates and Evolution	Highly scalable	Not very scalable	Static

2.2. Cyber ranges and adversary emulation

In recent years, the academic and industrial communities have developed various tools for adversary emulation. Frameworks like MITRE CALDERA and Red Canary's Atomic Red Team automate the execution of specific ATT&CK TTPs. However, these tools operate as endpoint agents and assume that the underlying network infrastructure already exists. To provide holistic training environments, educational "Cyber Ranges" such as SEED Labs and KYPO have been widely adopted. While historically reliant on heavy Virtual Machines (VMs), there has been a recent shift towards containerization to reduce hardware footprints. Despite this shift, standard Docker environments often lack native routing and Software-Defined Networking (SDN) capabilities required to emulate complex, multi-tier corporate topologies realistically. Our proposed Vuln_Box addresses this gap by using kathará, which bridges the lightweight nature of containers with advanced network emulation capabilities, providing a superior foundation for end-to-end behavioral threat modeling.

3. Proposed infrastructure: Vuln_Box architecture

The research utilizes Vuln_Box, a purpose-built virtualized environment that emulates a three-tier corporate network. The infrastructure is segmented into three security zones managed by a central router:

- Demilitarized Zone (DMZ): Hosts the perimeter-facing Tomcat server.
- Local Server Zone: Houses internal-facing web applications.
- Internal Network: Contains client workstations and nodes for vulnerability analysis.

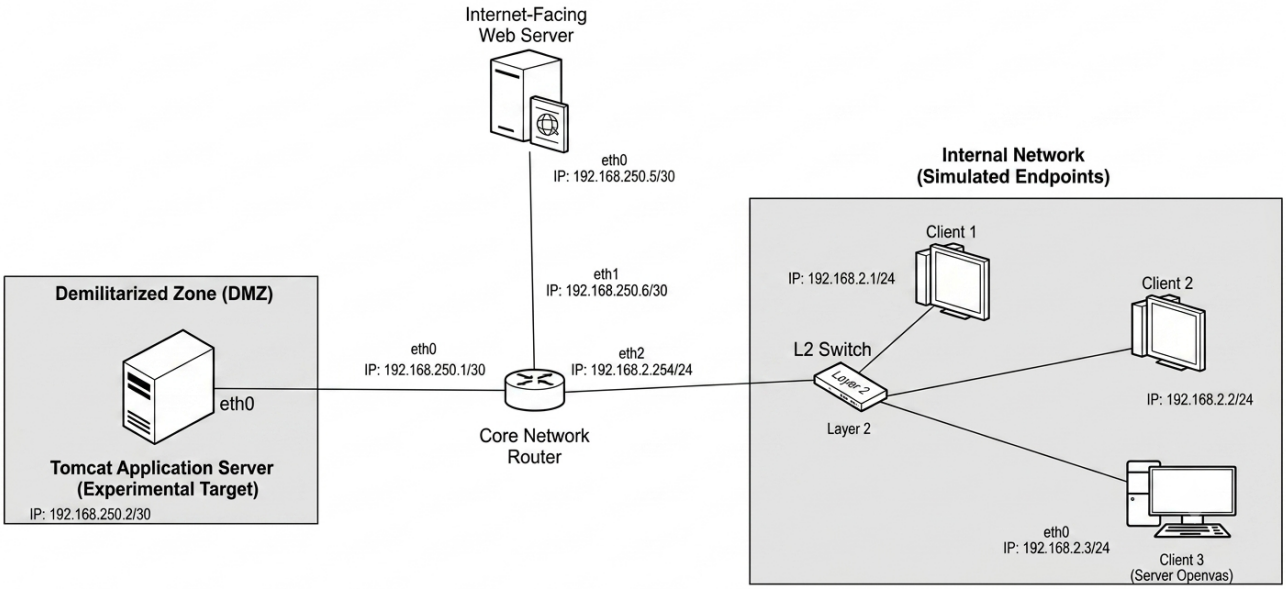

Figure 1: Vuln_Box Lab architecture

This segmentation allows a granular control of network flows and the simulation of multi-stage attack paths in real conditions by allowing us to attack the infrastructure from outside the network.

The implementation is done using the kathará network emulation engine [6]. By leveraging Docker-based containerization, kathará provides an "Infrastructure-as-Code" (IaC) framework where each node, including routers and servers, is a lightweight container. This allows the environment to be reset and re-deployed consistently for each experimental run.

Listing 1: Network configuration for the lab stored in the configuration file lab.conf

```
# PCs
linux1[num_terms]=1
pc2[num_terms]=0
server[num_terms]=1
webserver[num_terms]=1
tomcat[num_terms]=1

# Routers
r1[num_terms]=1

# Define devices in first zone (Internal)
linux1[0]="A"
r1[2]="A"
pc2[0]="A"
server[0]="A"
```

```
# Define devices in extranet zone
r1[1]="B"
webserver[0]="B"

# Define DMZ zone
r1[0]="D"
tomcat[0]="D"

# Docker Image Definitions for each node
tomcat[image]="tomcat"
linux1[image]="asgard-server"
```

4. Experimental setup and threat modeling methodology

We developed an attack scenario for the experiment that simulates a specific vulnerability. We explore the scenario by reproducing common bad practices that are observed as a context on the continent [9]. In fact, many recent cyberattacks, including those that were not publicly declared, have been in Africa. [15] With our scope of vulnerabilities being reduced, we only reproduced two server-side vulnerabilities considering their large impact and also the social engineering combination that is usually the most common for the second: *Spring4shell* and *target="_blank"*.
The adversarial modeling phase involves translating technical exploit executions into standardized Tactics, Techniques, and Procedures (TTPs) using the MITRE ATT&CK Matrix for Enterprise [5]. This process ensures that the results are not merely a report of system compromise but a behavioral analysis of the attack lifecycle.The methodology for selecting and mapping scenarios, such as Spring4Shell (CVE-2022-22965), follows a three-step systematic approach:

1. *Tool deployment*: While the infrastructure is automated via IaC (.startup and .conf files), the attack execution is now manual to mimic a real-world penetration test. Most of the tools we used are native tools while the Webshell (JSP) we exploited is delivered via an HTTP POST request that exploits the Module.getClassLoader() function in the vulnerable Spring Core Framework.[10]
2. *Exploit Decomposition*: The raw exploit is analyzed to identify its functional stages, such as the initial HTTP POST request targeting the DataBinder [11] and the subsequent web shell deployment.
3. *Technique Identification*: Each stage is assigned to specific ATT&CK IDs. We used the standard cyber kill chain approach to perform each attack. This translated into the use of nmap for scanning ports and services on targets and tools such as Gobuster and curl for reconnaissance and enumeration. The attacks also includes several other phases like: exploitation, collection and discovery, privilege escalation, and post-exploitation (See Table 2) the initial exploit targeting the vulnerable Spring application is classified under Exploit Public-Facing Application (T1190), while the command execution via the web shell is categorized as Command and Scripting Interpreter (T1059). [12] [13]
4. *Matrix Synthesis*: The observed results, whether successful exploitation or data or error leakages, are recorded to produce a curated matrix unique to the Vuln_Box environment. This mapping provides a granular view of the defensive gaps within the simulated infrastructure.

Table 2: Summary of tools used for infrastructure exploitation

Attack Phases	Used Tools	Objectives
Reconnaissance and Enumeration	Nmap, Gobuster, Curl	Identify accessible services and resources
Exploitation	spring4Shell-exploit, spring4Shell-scan, hydra	Execute remote code via the vulnerability
Collection and Discovery	Dcode, Bash	Search for sensitive files and useful information
Privilege Escalation	sudo, credential theft	Strengthen authentication with 2FA or MFA methods
Post-Exploitation and Persistence	WebShell (JSP)	Maintain remote access to the server

5. Results and analysis

5.1. Behavioral mapping results

In the context of the Vuln Box infrastructure, the relationship between Tactics, Techniques, and Procedures (TTPs) follows a hierarchical logic of adversarial intent:

- Tactics ("Why"): These represent the attacker's high-level strategic goals. In this case, Initial Access (TA0001) describes the goal of gaining a foothold in the DMZ.
- Techniques ("How"): These represent the specific method used to achieve a tactic. The targeting of the Spring4Shell vulnerability is categorized as exploiting a Public-Facing Application (T1190). A single tactic may involve multiple techniques (e.g., Discovery utilizes both File and Network service discovery).
- Procedures ("Action"): These are the specific, step-by-step implementations or tools used. For example, the procedure for T1190 in this study involved a crafted HTTP POST request to manipulate the ClassLoader of the target Java application.

Table 3: Mapping of Spring4Shell Attack Lifecycle to MITRE ATT&CK TTPs and Mitigations

Attack Phases	Tactic	Technique	Exploitation / Procedure	Mitigation
Reconnaissance and Enumeration	**Reconnaissance**	T1595, T1590	Nmap and curl used for service versioning and banner grabbing.	Disable banner grabbing; Restrict sensitive file access.
Exploitation	**Initial Access**	T1190	Malicious request to modify ClassLoader and drop a WebShell.	Update Spring Framework; Deploy WAF.
	Execution	T1059	JSP WebShell execution and abuse of authorized account permissions.	Restrict directory permissions; Deploy SIEM.
Privilege escalation	**Privilege Esc.**	T1068, T1548	Escalation via misconfigured SSH, FTP, and Tomcat root accounts.	Restrict root command execution; Strict ACLs.
	Defense Evasion	T1070	Deletion or modification of system logs to mask activity.	Centralized logging; Log integrity monitoring.

(Continued) Mapping of Spring4Shell Attack Lifecycle

Attack Phases	Tactic	Technique	Exploitation / Procedure	Mitigation
	Lateral Mov.	T1021	Pivoting to adjacent servers via stolen SSH/service credentials.	Network segmentation; Zero Trust architecture.
Collection and Discovery	**Credential Acc.**	T1552	Extraction of plaintext credentials from `credentials.txt`.	Use secure secrets managers; Encrypt all secrets.
	Discovery	T1083, T1046	Local and network enumeration using `ls`, `ps aux`, and `nmap`.	Implement Least Privilege Principle.
Post-Exploitation and Persistence	**Persistence**	T1505.003, T1098	Loading WebShell into `/webapps/ROOT/` for persistent access.	Enable IDS; Monitor suspicious HTTP requests.
	Impact	T1485	Final-stage deletion of critical files post-filtration.	Regular offline backups.

5.2. Infrastructure performance evaluation

To validate the "lightweight" claim of the Vuln_Box architecture, we evaluated its resource footprint against traditional virtual infrastructures including Kubernetes clusters and Virtual machines on Virtualbox. Because Kathará leverages Docker containerization through the Docker and Kubernetes (Megalos) manager [7], all emulated nodes (routers, DMZ servers, and internal clients) share the host operating system's kernel, eliminating the massive overhead of running multiple guest OSs. In addition, comparing it with solutions like Docker compose is not objective, since their core architecture is the same. We could easily compare it with a Kubernetes network though, but the Kubernetes orchestrator has a huge overhead in terms of resource cost making it impossible at this stage of our research to reproduce due to computing resources limitations. The remaining benchmark we can use for the comparison is Virtual infrastructure built from virtual machines. The strength of Kathará lies in three core aspects:

1. **Low resource consumption:** Compared to a Virtual infrastructure or a classic Kubernetes cluster, there is a fixed-cost elimination: Kathará pays per-device cost only (same as Compose/raw Docker), while K3s pays a control-plane tax on top of per-pod cost, even for a 3-node lab and virtualization uses most of the host's resources (memory and CPU included).4
2. **Fast deployment & Abstraction:** What is done manually with raw Docker commands for a 5-node with long .yaml files (e.g., 3-segment topology means dozens of sequential docker invocations with no topology-level validation), Kathará collapses that into a declarative file *.conf* file and a single command, with no additional runtime memory cost (the orchestration happens on the client-side, at lab-start time, not as a persistent daemon.[6]
3. **Optimized orchestration:** When you run *kathara lstart*, Kathará acts as an ephemeral CLI orchestrator rather than a persistent controller loop. This significantly reduces resource overhead and latency as kathará only needs to read the .conf file once to spawn all nodes in the default state [6].

Table 4: Performance Evaluation: Kathará vs. VirtualBox

Nodes	Strategy	Avg. CPU Load (%)	Avg. Free RAM (MB)
1 Node	Kathará	5.6%	4,131 MB
	VirtualBox	11.2%	534 MB
2 Nodes	Kathará	18.7%	3,796 MB
	VirtualBox	7.8%	283 MB
3 Nodes	Kathará	23.1%	3,475 MB
	VirtualBox	N/A*	N/A

Note: During our deployment of the 3-tier topology (comprising 5 nodes) on a host machine (8GB of RAM and 4 CPU Cores), the Kathará environment required an average of only 50–300 MB of RAM per node, with a total infrastructure (including the router used for interconnection) boot time of less than 15 seconds. In contrast, a traditional virtual machine setup would require at least 1–2 GB of RAM per node and several minutes to boot.

In addition, the simplicity of IaC and the commodity of.conf and .startup files under the Kathará environment also make it easier to use in any context for cheaper experimentation.

6. Discussion and operational impact

It appears important to mention the huge advantage that can be gained for security teams when they have behavioral based defense strategies, but then it also becomes a key point to use the right tool or framework to perform the threat modeling. The good practice is even to combine multiple frameworks, but we focused on MITRE ATT&CK because of its important adoption. This research clearly shows that by mapping successful exploits to the ATT&CK matrix, organizations can identify specific security gaps where techniques such as T1190 (Spring4Shell) are executed, but not detected by current log policies [14]. In addition, the infrastructure addresses the human element of cybersecurity. For junior analysts, the laboratory provides a safe, reproducible environment to witness the transition from a network alert to a multi-stage intrusion. This hands-on experience is essential for maturing an SOC from basic alert-monitoring to proactive Threat hunting.[16] These facets of the framework were key proofs of our research and are very promising if combined with AI based detection.
Although *Vuln_Box* provides a robust platform for behavioral analysis, certain technical constraints related to scope were identified during its implementation.

- **Emulation Scalability:** The use of kathará, while efficient, introduces complexity in large-scale node configurations. Startup script failures were observed in highly complex topologies, suggesting the need for more robust orchestration scripts in future iterations.
- **Temporal Gap in Threat Intelligence:** As a framework based on observed behaviors, MITRE ATT&CK is inherently retrospective. The current infrastructure effectively simulates known TTPs but lacks the predictive telemetry required to model *Zero-Day* vulnerabilities before they are codified into the framework.
- **Accessibility and Documentation:** To maximize the impact for the cybersecurity community in emerging markets, there is a recognized need for simplified open-source documentation to lower the barrier to laboratory deployment.

7. Conclusion and future work

This paper presented the design and implementation of Vuln_Box, a lightweight container-based infrastructure developed using the Kathará emulation engine to facilitate behavioral threat analysis. By successfully mapping the multi-stage exploitation of vulnerabilities such as Spring4Shell (CVE-2022-22965) to the MITRE ATT&CK framework, this work demonstrates how architectural emulation can

bridge the gap between theoretical vulnerability assessment and threat intelligence operations. The results provide a granular TTP mapping that enhances the capabilities and possibilities of defensive gap analysis and serves as a training ground for junior security analysts.
Future research will focus on evolving the infrastructure from a static lab to a proactive defense platform. A primary objective is the integration of Agentic AI to facilitate autonomous threat hunting with a focus on the threat landscape in the financial sector in West African microfinance institutions . By deploying intelligent agents within the Vuln_Box environment, we aim to automate the detection of non-linear adversarial patterns and develop self-healing network protocols. Such advances will significantly improve overall security and also build the foundations for AI-driven resilient infrastructure capable of countering modern cyber threats.

A. Appendices

To replicate the infrastructure, we have provided a repository where all the startup files and dockerfiles for the OS deployed on the nodes are stored.

A.1. Repository and Source Code Availability

The complete source code, Infrastructure-as-Code (IaC) configuration files, network orchestration scripts, and laboratory documentation for the Vuln_Box project are openly accessible in the public GitHub repository:

https://github.com/w4lk3r04/internship-projects/tree/vuln_lab

B. Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini to rephrase sentences or paragraphs to improve clarity and conciseness in the sections: Experimental setup and threat modeling methodology and Results and analysis. After using this tool, the author(s) reviewed and edited the content as needed and assume(s) full responsibility for the content of the publication.

References

[1] Microsoft, "Microsoft Digital Defense Report 2024: Les cyberattaques ont presque doublé d'une année sur l'autre," 2024. [Online]. Available: https://news.microsoft.com/source/emea/2024/10/microsoft-digital-defense-report-2024-les-cyberattaques-ont-presque-double-dune-annee-sur-lautre/

[2] K. Kim, I. M. Alshenaifi, S. Ramachandran, J. Kim, T. Zia, and A. Almorjan, "Cybersecurity and Cyber Forensics for Smart Cities: A Comprehensive Literature Review and Survey," *Sensors*, vol. 23, no. 7, p. 3681, Apr. 2023. doi: 10.3390/s23073681.

[3] S. Zubair, M. Zubair, and M. Ahmed, "Investigation on the Infusion of Cybersecurity and Smart City," in *Cybersecurity for Smart Cities: Practices and Challenges*, M. Ahmed and P. Haskell-Dowland, Eds. Cham: Springer, 2023, pp. 1–15. doi: 10.1007/978-3-031-24946-4. [Online]. Available: https://doi.org/10.1007/978-3-031-24946-4.

[4] "What is the Mitre Att&ck Framework? | CrowdStrike," CrowdStrike.com. Accessed: Jun. 14, 2026. [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/mitre-attack-framework/

[5] MITRE, "MITRE ATT&CK Framework - Enterprise Techniques," 2024. [Online]. Available: https://attack.mitre.org/techniques/enterprise/

[6] M. Scazzariello, L. Caiazzo, and T. Ariemma, "Kathará: A lightweight network emulation system," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, IEEE, 2020, pp. 1–2. doi: 10.1109/NOMS47738.2020.9110311.

[7] M. Scazzariello, L. Ariemma, G. D. Battista, and M. Patrignani, "Megalos: A Scalable Architecture for the Virtualization of Network Scenarios," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, 2020, pp. 1–7, doi: 10.1109/NOMS47738.2020.9110288.

[8] B. Al-Sada, A. Sadighian, and G. Oligeri, "MITRE ATT&CK: State of the Art and Way Forward," *ACM Computing Surveys*, vol. 57, no. 1, Art. 12, 2024. doi: 10.1145/3687300.

[9] Francois P. Cornelius and Shandre K. Jansen van Rensburg. Emerging South African smart cities: Data security and privacy risks and challenges. *South African Journal of Information Management*, 26(1):a1847, 2024. Available at: https://sajim.co.za

[10] Z. A. Salem Frederic Baguelin, Emile Spir, Eslam, 'The Spring4Shell vulnerability: Overview, detection, and remediation | Datadog Security Labs'. Accessed: Jun. 13, 2026. [Online]. Available: https://securitylabs.datadoghq.com/articles/spring4shell-vulnerability-overview-and-remediation/

[11] Rapid7, "Detect the Spring4Shell vulnerability | Vulnerability Management Documentation," *Rapid7 InsightVM Documentation*, Mar. 13, 2026. [Online]. Available: https://docs.rapid7.com/insightvm/spring4shell/. [Accessed: Mar. 13, 2026].

[12] MITRE ATT&CK, "Exploit Public-Facing Application (T1190)," *MITRE ATT&CK Framework*, 2024. [Online]. Available: https://attack.mitre.org/techniques/T1190/. [Accessed: Mar. 13, 2026].

[13] MITRE ATT&CK, "Command and Scripting Interpreter (T1059)," *MITRE ATT&CK Framework*, 2024. [Online]. Available: https://attack.mitre.org/techniques/T1059/. [Accessed: Mar. 13, 2026]

[14] Roy, S., Panaousis, E., Noakes, C., Laszka, A., Panda, S., and Loukas, G. (2023). SoK: The MITRE ATT&CK Framework in Research and Practice. *arXiv preprint arXiv:2304.07411*. https://doi.org/10.48550/arXiv.2304.07411

[15] Kaspersky. (2025). *Africa Cyberthreat Landscape Report 2025*. Retrieved January 30, 2026, from https://content.kaspersky-labs.com/se/media/en/africa-cyberthreat-landscape-report-2025.pdf

[16] Alnajim, A.M., Habib, S., Islam, M., AlRawashdeh, H.S., and Wasim, M. (2023). Exploring Cybersecurity Education and Training Techniques: A Comprehensive Review of Traditional, Virtual Reality, and Augmented Reality Approaches. *Symmetry*, 15(12), 2175. https://doi.org/10.3390/sym15122175