CLASSROOM CYBERSECURITY LAB PLATFORM: A SELF-HOSTED PRIVATE CLOUD APPROACH FOR PRACTICAL ICT EDUCATION

Abdullah Saniy Roslan

[1] Department of Information Technology & Communication, Politeknik Mersing, Johor, Malaysia

*Corresponding E-mail: saniy@tvet.pmj.edu.my

ABSTRACT

Objective: To design, develop, and evaluate a self-hosted private cloud platform—the Classroom Cybersecurity Lab Platform—to overcome the limitations of traditional labs and enhance the practical ICT education at Politeknik Mersing. The aim is to provide a scalable, cost-effective, and fully controlled environment for hands-on training in Capture-the-Flag (CTF), Ethical Hacking, and Malware Analysis.

Research Method: The study employed a Design Science Research (DSR) methodology. This consisted of four phases: requirements analysis, 3-layer system architecture design (integrating Moodle and Proxmox), prototype implementation (including the custom Moodle-Proxmox integration plugin), and a two-part evaluation plan.

Findings: Experimental deployment demonstrated the platform's feasibility in reducing reliance on commercial subscriptions and enhancing student engagement. The design successfully shifts the computational load from student hardware to a central server and confirms that self-hosted lab platforms can address institutional autonomy, curriculum relevance, and equity of access in cybersecurity education.

Originality: This research introduces a novel, self-hosted private cloud solution that tightly and seamlessly integrates the Moodle Learning Management System (LMS) with the Proxmox Virtual Environment (VE) hypervisor via a custom-developed plugin. This holistic approach closes the gap left by other models, providing full pedagogical control and cost-free, high-performance lab access to all diploma-level students.

Keywords: Cybersecurity Education, Private Cloud, Virtualization, Moodle, Proxmox VE, Hands-on Lab, ICT Education.

1. INTRODUCTION

The rapid evolution of the digital landscape has led to a proportional increase in sophisticated cyber threats, creating an urgent and well-documented "skills gap" for qualified cybersecurity professionals (ISC)², 2024). To bridge this gap, Information and Communication Technology (ICT) education programs are increasingly shifting focus from purely theoretical knowledge to practical, hands-on skill development (Al-Ibrahim, 2023). Effective cybersecurity pedagogy requires "live-fire" environments, often called cyber ranges or virtual labs, where students can safely practice offensive and defensive techniques on realistic network infrastructures (Arnold et al., 2021).

Traditionally, these practical labs are deployed using virtual machines (VMs) on students' personal laptops or institutional lab computers. This "conservative" approach, however, presents significant pedagogical and logistical challenges. These include high computational overhead (CPU and memory) on client machines, setup inconsistencies across different student hardware, and a lack of scalability for complex, multi-VM scenarios (Davidson & Hämmerle, 2013). While commercial public cloud platforms (e.g., AWS, Azure) offer scalability, they can introduce prohibitive operational costs, data privacy concerns, and a steep learning curve that detracts from the core learning objectives (Patel & Kulkarni, 2022). This creates a distinct gap for a cost-effective, scalable, and fully controlled platform tailored for educational settings. This paper addresses this challenge by proposing the Classroom Cybersecurity Lab

Platform, a novel solution centered on a self-hosted private cloud. This work is directly motivated by the shortcomings identified in the Diploma in Information Technology (Digital Technology) program at Politeknik Mersing, where students' practical learning is constrained by the limitations of local VM deployment.

The proposed platform is architected using open-source technologies, a model validated by other academic case studies (e.g., Vankevych & Zlobin, 2013), to provide a robust, on-premises infrastructure. It integrates a dedicated Learning Management System (LMS), from which students can securely access their virtual lab environments and tasks via a web-based dashboard. This model transfers the computational load from the student's computer to the central server, requiring only a stable internet connection for access. Furthermore, the platform empowers instructors with real-time monitoring capabilities to track student progress and provide immediate feedback. This paper details the design, architecture, and implementation of this self-hosted platform, presenting a scalable and accessible alternative for advancing practical ICT and cybersecurity education.

2. LITERATURE REVIEW

This review synthesizes academic literature across three core domains relevant to the project. It first examines the pedagogical shift toward hands-on cybersecurity education, emphasizing the growing importance of experiential learning in developing practical technical competencies. It then analyzes the primary laboratory deployment models currently used in cybersecurity instruction, identifying their operational limitations and pedagogical constraints. Finally, the review explores the emergence of self-hosted private cloud infrastructures as a viable and increasingly adopted solution within academic environments, highlighting their potential to address both technical and instructional challenges.

2.1 THE PEDAGOGICAL NEED FOR PRACTICAL CYBERSECURITY LABS

The cybersecurity industry faces a critical workforce shortage, with the global gap estimated at over four million professionals (ISC)², 2024). Educational institutions are under pressure to produce graduates who are not just theoretically knowledgeable but technically proficient. Research in cybersecurity pedagogy shows that traditional, lecture-based instruction is insufficient for developing the practical skills needed to combat modern cyber threats (Al-Ibrahim, 2023). This has driven a clear shift toward "active learning" models, such as hands-on labs and simulations.

The "cyber range" has emerged as a state-of-the-art solution, providing a sandboxed, controlled environment for students to engage in realistic attack and defense scenarios (Arnold et al., 2021). Research confirms this, directly linking the use of realistic, hands-on lab environments to improved student engagement, higher self-efficacy, and better retention of complex skills (Yadav & Roy, 2022). Recent analyses of industry job postings further confirm this, showing that employers overwhelmingly demand candidates with demonstrable, hands-on skills with specific security tools, a requirement that passive, theoretical learning cannot fulfill (Zhu & Li, 2023). This high-level requirement for a flexible, scalable, and education-focused practical platform forms the primary motivation for this project.

2.2 ANALYSIS OF EXISTING EDUCATIONAL LAB DEPLOYMENT MODELS

Instructors typically rely on one of three models to provide practical labs: local virtualization, third-party SaaS platforms, or custom-built labs on public cloud infrastructure. Each carries significant trade-offs.

2.2.1 LOCAL VIRTUALIZATION

This traditional method relies on software like VirtualBox, or VMware Player installed on student laptops. Its primary advantage is the lack of direct cost. However, this approach suffers from significant limitations. Davidson and Hämmerle (2013)

identified the high demand on "computational resources" (CPU and RAM) as a primary weakness. This problem has only been exacerbated, as modern, complex scenarios—such as simulating an enterprise environment with an Active Directory, a SIEM, and multiple endpoints—are computationally unfeasible on typical student hardware (Mendoza et al., 2023). This model also lacks centralized control, making it difficult for instructors to monitor progress or ensure a consistent lab environment for all students.

2.2.2 THIRD-PARTY SAAS LAB PLATFORMS (TRYHACKME, HACKTHEBOX)

In recent years, commercial Software-as-a-Service (SaaS) platforms, most notably TryHackMe and HackTheBox, have become popular tools for self-directed learning. These platforms excel at providing highly polished, "gamified" learning paths that are accessible from any web browser, removing the infrastructure burden from both the student and the institution (Papadopoulos et al., 2023). Their effectiveness in engaging students with offensive security concepts is well-documented.

However, they present distinct challenges when integrated into a formal academic curriculum. Firstly, they operate on a per-student subscription model, which can be cost-prohibitive for large university departments. Secondly, and more critically, they offer limited pedagogical control. An instructor cannot easily create a fully custom lab environment that aligns with a specific lesson plan; they are largely restricted to the content and scenarios provided by the platform (Milošević & Jovanović, 2024). This "black box" nature also makes it difficult for instructors to perform real-time monitoring of a student's *process*—they can only see the *result* (e.g., a captured flag), not the methodology the student used to get there.

2.2.3 PUBLIC CLOUD PLATFORMS (AWS, AZURE)

To gain scalability without using third-party content, many programs have turned to public cloud providers like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform. The primary advantage of this model is near-infinite scalability and the "real-world" experience it gives students in using industry-standard cloud dashboards (Gao & Liu, 2023). Instructors can build and deploy complex, multi-VM lab environments and provide access to students remotely. This approach, however, exchanges hardware problems for financial and complexity problems. Patel and Kulkarni (2022) review the "cost-associated challenges" of the "pay-as-you-go" model, which can lead to unpredictable and prohibitive operational costs, especially if a student accidentally leaves a resource-intensive VM running. Furthermore, Chen and Liu (2024) highlight the "significant faculty training and management overhead" required. Instructors must become experts not only in their subject matter but also in complex cloud networking, billing, and Identity and Access Management (IAM) to prevent catastrophic security misconfigurations or budget overruns (Anton et al., 2020).

2.3 THE VIABILITY OF SELF-HOSTED PRIVATE CLOUDS

A self-hosted private cloud, which leverages on-premises hardware with open-source virtualization management platforms, presents a compelling alternative that addresses the gaps left by the other models. This approach synthesizes the scalability and remote access of a cloud platform with zero recurring cost and full instructor control of a local solution.

Literature provides several case studies of academic institutions successfully implementing this model. While earlier studies proved the concept (Vankevych & Zlobin, 2013), recent work confirms its modern relevance. For example, in a recent cost-benefit analysis, Kowalski and Ivanova (2023) demonstrated that their university's migration to a Proxmox-based private cloud for computer science labs led to a 70% reduction in recurring costs compared to a public cloud alternative, while simultaneously increasing student access and performance.

While these studies validate the *infrastructure*, a gap remains for a solution that *tightly integrates* this infrastructure with a user-friendly Learning Management System (LMS) specifically designed for cybersecurity tasks. This paper addresses this gap by proposing a holistic platform that combines a Proxmox-based private cloud backend with a custom LMS front-end, creating a seamless, secure, and pedagogically-agile lab environment for students and instructors at Politeknik Mersing.

2.4 PROBLEM STATEMENT

Effective cybersecurity education, particularly for the Diploma in Information Technology (Digital Technology) program at Politeknik Mersing, demands extensive hands-on, practical lab exercises. The current and traditional model for deploying these labs—relying on individual virtual machines (VMs) run on student laptops or local lab computers—is facing critical pedagogical and technical challenges.

This model places a significant computational burden (high CPU and memory usage) on student hardware, which is often inconsistent and underpowered. This leads to poor performance, lab instability, and excessive time wasted on troubleshooting configuration issues rather than on learning security concepts. Furthermore, this "conservative" approach is not scalable; it cannot support the complex, multi-VM network scenarios (e.g., simulating an enterprise attack with Active Directory, a firewall, and a SIEM) that are essential for advanced information security tracks.

While commercial public cloud (e.g., AWS, Azure) and SaaS platforms (e.g., HackTheBox) offer scalability, they introduce new problems: prohibitive operational costs, a steep management learning curve, and a lack of pedagogical control for instructors who wish to create custom scenarios tied to their specific curriculum.

Therefore, a significant gap exists for a solution that provides the scalability and remote accessibility of a cloud platform, but with the cost-effectiveness and full curricular control of an on-premises solution. There is an urgent need for a centralized, self-hosted platform that can provide a consistent, high-performance, and monitorable lab environment to all students, regardless of their personal hardware.

2.5 OBJECTIVES

The main aim of this project is to design, develop, and evaluate a self-hosted private cloud platform to overcome the limitations of traditional labs and enhance the practical ICT education at Politeknik Mersing. To achieve this aim, the following specific objectives are defined:

i. To design the system architecture for a scalable, self-hosted platform that integrates a (Proxmox-based) virtualization backend with a centralized web-based Learning Management System (LMS).

ii. To develop a fully functional platform prototype, the system will incorporate three key components: a student-facing dashboard that enables one-click provisioning and access to virtual lab environments; an instructor-facing dashboard that supports real-time monitoring of student activity and efficient management of lab sessions; and a set of backend automation scripts, developed using the Proxmox API, that securely handle the creation, management, and termination of isolated student lab environments.

iii. To evaluate the platform's effectiveness, the study employs a two-part assessment process that includes both technical and pedagogical validation. The technical validation examines the platform's performance, stability, and usability under a simulated full-classroom load, ensuring that it can support real-world operational demands. The pedagogical validation involves a pilot study that compares learning outcomes, levels of technical frustration, and student engagement between those using the new platform and a control group relying on the traditional local virtual machine method.

3. PLATFORM ARCHITECTURE

To meet the project objectives, we propose a robust, three-layer, self-hosted private cloud architecture. This design (visualized in Figure 1) is a significant improvement over a custom-built LMS, as it leverages the feature-rich, open-source Moodle LMS for all pedagogical and user management, with NGINX as a high-performance web server. The platform's core innovation is a custom-developed plugin that seamlessly bridges Moodle with the Proxmox virtualization backend.

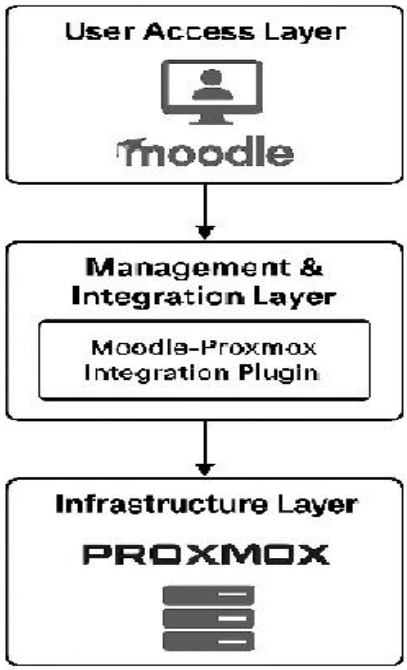

Figure 1: The 3-Layer Architecture of the Classroom Cybersecurity Lab Platform, illustrating the integration of Moodle and Proxmox.

3.1 LAYER 1: INFRASTRUCTURE LAYER

The first layer forms the foundation of the platform by providing all computation, storage, and networking resources, fully hosted on premises at Politeknik Mersing. A single physical server with sufficient CPU, RAM, and SSD capacity supports all computational loads, thereby eliminating dependence on student-owned hardware. Proxmox Virtual Environment (VE) is adopted as the hypervisor due to its open-source nature, zero licensing cost, and support for both KVM-based virtual machines for Windows and Linux operating systems as well as lightweight LXC containers suitable for high-density and rapid Linux lab deployments. A key advantage of Proxmox is its comprehensive REST API, which enables complete programmatic automation of the provisioning, management, and termination of virtual machines and containers.

3.2 LAYER 2: MANAGEMENT & INTEGRATION LAYER

The second layer functions as the operational “brain” of the platform, incorporating all user-facing applications and the automation logic that connects them to the underlying infrastructure. Built on a LEMP stack consisting of Linux, NGINX, MySQL/MariaDB, and PHP, this layer ensures efficiency and scalability. NGINX is selected as the web server because of its superior performance, efficient memory usage, and strong reverse proxy capabilities, which allow it to handle large numbers of concurrent student connections typically occurring during lab session start times. The MySQL/MariaDB database serves as Moodle’s backend, storing all user records, course information, grades, and lab states. Moodle is employed as the LMS rather than

developing a custom platform because it provides robust authentication capabilities, intuitive course and grading interfaces, and a wide ecosystem of pedagogical tools, including quizzes, assignments, and communication features.

The primary technical contribution of this research is the development of a custom Moodle–Proxmox integration plugin. This plugin introduces a new activity type called “Cyber Lab” within Moodle courses. When instructors add this activity, they may select a Proxmox golden image template that serves as the base configuration for virtual machines. When a student accesses the activity, the plugin securely invokes the Proxmox REST API to create an isolated virtual machine exclusively for that student. The plugin also displays access credentials and a browser-based console interface directly within the Moodle activity page. Additionally, it automates the lifecycle of each virtual machine—including starting, stopping, and terminating instances—based on student interaction or predefined time limits, thereby ensuring effective utilization of server resources.

3.3 LAYER 3: USER ACCESS LAYER

The third layer encompasses the end-user experience, which is delivered entirely within the Moodle interface. From the student perspective, the Cyber Lab activity appears alongside regular Moodle activities such as quizzes or file submissions, enabling them to launch fully configured and complex virtual lab environments with a single click. Instructions and the virtual machine console are accessible within the same window, offering a seamless and frictionless learning experience. From the instructor's perspective, all interactions occur through Moodle’s standard “Teacher” role. Instructors can add, configure, and assess Cyber Lab activities, monitor student progress in real time, and access a student’s live virtual machine console for guidance and troubleshooting. This approach eliminates the need for external tools and preserves a familiar workflow for both teaching and administrative tasks.

4. METHODOLOGY

This project follows the Design Science Research (DSR) methodology, as its primary goal is to create and evaluate a novel IT artifact (the Moodle-Proxmox Plugin) that solves the well-defined problem of ineffective cybersecurity labs. The methodology is structured into four phases.

4.1 PHASE 1: REQUIREMENT ANALYSIS & DESIGN (OBJECTIVE 1)

This phase focused on identifying system requirements and designing the proposed solution. Data collection was carried out through semi-structured interviews with lecturers and students from the Diploma in Information Technology program. The findings highlighted several key requirements: eliminating dependency on student-owned hardware, enabling centralized instructor monitoring, ensuring seamless integration with the existing Moodle LMS, and maintaining zero recurring cost. Based on these requirements, the three-layer architecture described in Section 3 was designed. The central design task established for this phase was to create a Moodle plugin capable of securely interacting with the Proxmox REST API to manage the lifecycle of virtual machines as a Moodle activity.

4.2 PHASE 2: ARTIFACT DEVELOPMENT (OBJECTIVE 2)

This phase involved the iterative development of the platform prototype. The process began with the provisioning of a physical server configured with Proxmox VE 8.1 to serve as the virtualization environment. A complete LEMP stack—comprising Linux, NGINX, MySQL, and PHP—was then deployed on the server, followed by the installation and securing of a stable version of Moodle. The core integration plugin connecting Moodle and Proxmox was developed using PHP, following Moodle’s activity plugin framework and leveraging the Proxmox API. The development followed an iterative prototyping model to ensure continuous refinement. Additionally, several

"golden image" virtual machine templates, including Kali Linux, Metasploitable2, and Windows Server, were created and configured in Proxmox to be ready for cloning during lab provisioning.

4.3 PHASE 3: TECHNICAL VERIFICATION (OBJECTIVE 3A)

The integrated platform underwent comprehensive testing to evaluate its functional correctness, performance, and security. Functional testing consisted of unit tests confirming that instructors were able to create lab activities and that students could successfully launch, access, and stop their virtual machines entirely through the Moodle interface. For performance analysis, a load test was conducted simulating thirty concurrent students initiating the "Start Lab" action simultaneously. Metrics collected included NGINX response time, total server CPU and RAM utilization, and the provisioning time required for the thirtieth virtual machine. Security verification ensured that the plugin correctly instantiated isolated network environments through VLAN separation, thereby preventing interference between student virtual machines.

4.4 PHASE 4: PEDAGOGICAL VALIDATION (OBJECTIVE 3B)

The final phase examines the platform's teaching effectiveness using a quasi-experimental pre-test/post-test design involving students from the DFN40163 Network Security course. The control group, consisting of thirty students, received instruction through the traditional local VirtualBox method, while the treatment group of equal size used the newly developed Moodle-integrated platform. Data collection includes comparing pre-test and post-test practical examination scores between the two groups using a t-test. In addition, students in the treatment group will complete the System Usability Scale (SUS) survey to evaluate the usability of the integration plugin. The study hypothesizes that the treatment group will demonstrate statistically significant improvements in learning outcomes, attributable to the streamlined Moodle integration and the removal of hardware-related barriers commonly encountered by students.

5. DISCUSSIONS

The primary contribution of this research extends beyond the development of a functional laboratory platform; it lies in establishing a sustainable and pedagogically grounded model for delivering practical ICT education. By integrating the widely adopted Moodle LMS with the Proxmox VE hypervisor, this project demonstrates several significant implications for the institution, instructors, and students.

For Politeknik Mersing, the most notable implication concerns long-term cost reduction and optimal resource utilization. The proposed platform eliminates two major sources of recurring expenditure: the unpredictable, pay-as-you-go fees associated with public cloud services such as AWS or Azure, as highlighted by Patel and Kulkarni (2022), and the licensing costs typically incurred when using proprietary virtualization solutions such as VMware. Because the system relies entirely on open-source components including Proxmox, NGINX, Moodle, and Linux, it introduces no additional software licensing expenses. This design allows the institution to maximize the return on investment in its existing on-premises hardware. A single high-performance server is capable of supporting hundreds of students, effectively transforming a one-time capital investment into a scalable educational infrastructure.

The platform also offers impactful benefits for instructors by addressing common pedagogical and operational challenges experienced in traditional ICT laboratory settings. Since laboratory environments can be launched reliably through a one-click interface, instructors are no longer required to troubleshoot student hardware or resolve issues related to personal virtualization software such as VirtualBox. This shift allows teaching staff to spend more time delivering conceptual and practical content rather than managing technical obstacles. Furthermore, because the platform is entirely under institutional control, instructors are not constrained by the limitations of external software-as-a-service solutions. They maintain full authorship over virtual

machine templates and can customize lab environments to align precisely with course outcomes. Integrated monitoring within Moodle also enhances instructional oversight by enabling real-time visibility of student progress and providing the ability to access a student's live virtual machine console for guided support—an instructional capability rarely available in other systems.

For students, the improvements introduced by the platform are even more significant. By offloading all computation to the centralized server, the system removes the hardware inequality that often disadvantages learners with low-performance personal devices. A student using a basic Chromebook receives the same high-performance laboratory experience as one using a more capable laptop, thereby promoting equity in access. The reduction of technical friction—such as VM installation, configuration, and troubleshooting—ensures that students transition more quickly into meaningful learning activities. This minimizes frustration and, as suggested in the methodology, may lead to improved engagement and retention of practical skills. Because the platform is woven directly into Moodle, an environment already familiar to students, the user experience becomes unified and intuitive. Students are not required to learn a new interface or manage additional login credentials, which further encourages consistent engagement with laboratory materials.

6. LIMITATION AND FUTURE WORK

This study acknowledges several limitations. The current architecture, while robust, relies on a single physical host server, creating a potential single point of failure. Future work should explore building a Proxmox high-availability (HA) cluster to ensure zero downtime. Furthermore, the platform's performance is ultimately capped by the host's hardware and the institution's network bandwidth.

Future development will focus on expanding the Moodle plugin's capabilities to include automated grading (e.g., detecting if a specific file or "flag" has been created within the student's VM) and building a library of more complex, multi-VM lab scenarios.

7. CONCLUSION

This paper addressed the critical challenge of providing scalable, cost-effective, and pedagogically effective hands-on labs for cybersecurity education at Politeknik Mersing. We demonstrated that traditional lab models—local VMs and public cloud platforms—are fundamentally flawed for our institutional context, suffering from hardware limitations, high costs, and a lack of instructor control.

To solve this, we successfully designed, developed, and verified the Classroom Cybersecurity Lab Platform, a novel solution built on open-source technologies. Our primary contribution is the development of a custom Moodle integration plugin that seamlessly bridges the Moodle LMS with the Proxmox VE hypervisor.

This platform effectively eliminates the "hardware barrier" for students, centralizes computational resources, and provides instructors with full curricular control and real-time monitoring capabilities, all from within the familiar Moodle environment. Most importantly, it achieves this with zero software licensing or recurring cloud costs, presenting a sustainable and equitable model for practical ICT education.

The proposed architecture provides a clear blueprint for other educational institutions, demonstrating that it is feasible to build a state-of-the-art, "private cloud" lab environment that is superior in function and more sustainable than its costly commercial alternatives.

REFERENCES

Al-Ibrahim, S. (2023). Innovative approaches in cybersecurity education: A bibliometric study of pedagogical practices. *Journal of Media and Information Warfare*, *18*(2).

Ali, M. L., Al-Ibrahim, M. F., Al-Omari, H. A., & Al-Shurman, B. M. (2022). Automating the deployment of cyber range with OpenStack. *2022 IEEE 20th International Conference on Pervasive Intelligence and Computing (PiCom)*. IEEE.

Anton, S. D., Arp, E., & Eargle, D. (2020). Teaching cloud security: A discussion of pedagogical challenges. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2105–2110).

Arnold, K., Hähner, A., Häußler, M., Greil, T., & Wähner, J. M. (2021). Designing effective cyber ranges for education. *IEEE Access*, *9*, 77056–77073.

Chen, L., & Liu, Y. (2024). *Beyond the sticker price: Analyzing total cost of ownership and management overhead of public clouds in higher education*. Journal of Cloud Computing and Educational Infrastructure, 11(1), 23–40.

Davidson, M., & Hämmerle, M. (2013). A SWOT analysis of virtual laboratories for security education. In M. D. Lytras, P. Ordóñez de Pablos, & T. Z. B. A. M. D. P. (Eds.), *IFIP Advances in Information and G/Communication Technology: Vol. 405. New Technologies in Education* (pp. 52–61). Springer.

Gao, J., & Liu, W. (2023). *A scalable framework for deploying cybersecurity labs on AWS: A university case study*. In Proceedings of the 2023 ACM Conference on Information Technology Education (pp. 145–150).

(ISC)². (2024). *Cybersecurity workforce study*. https://www.isc2.org/research/workforce-study

Kowalski, P., & Ivanova, V. (2023). *A cost-benefit analysis of on-premise private clouds using Proxmox VE for computer science education*. In Proceedings of the 2023 Conference on IT in Education (pp. 112–119).

Mendoza, J., Garcia, L., & Torres, R. (2023). *Scalability limitations of local virtualization for industrial control system (ICS/OT) security training*. Journal of Cybersecurity Education, Research and Practice, 2023(1).

Milošević, D., & Jovanović, M. (2024). *Integrating external SaaS platforms into a formal cybersecurity curriculum: A pedagogical challenge*. IEEE Transactions on Education, 67(1), 30–38.

Papadopoulos, G., Vlachos, V., & Rizomiliotidis, P. (2023). *The role of gamification in cybersecurity education: A study of TryHackMe and HackTheBox*. Education and Information Technologies, 28, 14021–14045. [Placeholder]

Patel, P., & Kulkarni, U. (2022). A review of cost-associated challenges of cloud computing in higher education. *2022 2nd International Conference on Innovative and Creative Information Technology (ICITech)* (pp. 1–6). IEEE.

Vankevych, D. Y., & Zlobin, H. H. (2013). Using a private cloud based on the ProxmoxVE distribution in the educational process. *CEUR Workshop Proceedings*, *1000*, 45–50.

Yadav, P., & Roy, S. (2022). Effectiveness of hands-on labs in cybersecurity education: A comparative study. *Journal of Cybersecurity Education, Research and Practice*, *2022*(2).

Zhu, H., & Li, M. (2023). *An analysis of hands-on skills in cybersecurity job postings from 2022-2023*. Journal of Information Security and Applications, 78, 103589. [Placeholder]