SIEM-as-a-Service for Private Clouds: Operational Characterization of a Wazuh Deployment Under SSH Brute-Force Workloads

Edivaldo Pastori Valentini[1,2], **Lilia Gomes de Matos**[1], **Gustavo A. Tuchlinowicz Nunes**[1], **Matheus Goulart Ranzani**[1], **Gabriel de Avelar Las Casas Rebelo**[1], **Rodolfo Ipolito Meneguette**[1], **Júlio Cezar Estrella**[1]

[1] Institute of Mathematics and Computer Sciences (ICMC)
University of São Paulo (USP)
São Carlos – SP – Brazil

[2]Federal Institute of São Paulo (IFSP) – Campus Catanduva
Catanduva – SP – Brazil

edivaldopv@usp.br, liliamatos@usp.br
gustavoalexandre@usp.br, matheus.ranzani@usp.br
gabrieldeavelar@usp.br
meneguette@icmc.usp.br, jcezar@icmc.usp.br

Abstract. *The increasing sophistication of cyberattacks and the growing operational complexity of private clouds have intensified the need for continuous and reliable monitoring of distributed services. In such environments, effective security event management depends on the timely collection, transport, and correlation of heterogeneous logs across multiple hosts and layers, including virtual machines, containers, and networked services. This paper presents a reproducible SIEM-as-a-Service deployment for private clouds based on the Wazuh platform, containerized with Docker Compose and instrumented with Prometheus and Grafana to provide operational observability.*
To stress the ingestion and correlation pipeline under realistic adversarial conditions, we generate distributed SSH brute-force attacks using concurrent attackers and bursty inter-arrival times modeled by an exponential distribution. The system is evaluated under a multifactor experimental design that varies both attack intensity and server resource allocation, enabling a systematic analysis of alert generation behavior and operational metrics, including CPU utilization and alert volume.
Results obtained from nine experimental scenarios indicate that the configuration with 8 vCPUs and 16 GB of RAM provides the most stable behavior, combining high alerting capacity with regular and predictable resource utilization. These findings offer a practical sizing baseline for on-premises SIEM deployments and highlight the importance of provisioning based on peak adversarial conditions rather than average workload.

Resumo. *A crescente sofisticação dos ciberataques e a complexidade operacional cada vez maior das infraestruturas de nuvem privada intensificaram a necessidade de monitoramento contínuo e confiável de serviços distribuídos. Nesses*

ambientes, o gerenciamento eficaz de eventos de segurança depende da coleta, transporte e correlação oportunos de logs heterogêneos em múltiplos hosts e camadas, incluindo máquinas virtuais, contêineres e serviços em rede. Este artigo apresenta uma implementação reproduzível de SIEM como serviço para nuvens privadas baseada na plataforma Wazuh, conteinerizada com Docker Compose e instrumentada com Prometheus e Grafana para fornecer observabilidade operacional de ponta a ponta. Para testar a capacidade do pipeline de ingestão e correlação sob condições adversárias realistas, geramos ataques de força bruta distribuídos via SSH usando atacantes simultâneos e intervalos de tempo entre chegadas variáveis, modelados por um processo exponencial. O sistema é avaliado sob um projeto experimental multifatorial que varia tanto a intensidade do ataque quanto a alocação de recursos do servidor, permitindo uma análise sistemática do comportamento de geração de alertas e das métricas operacionais, incluindo utilização da CPU e volume de alertas. Os resultados obtidos em nove cenários experimentais indicam que a configuração com 8 vCPUs e 16 GB de RAM proporciona o comportamento mais estável, combinando alta capacidade de alerta com utilização de recursos regular e previsível. Essas descobertas oferecem uma base prática para dimensionamento de implantações SIEM locais e destacam a importância do provisionamento com base em condições adversas máximas, em vez da carga de trabalho média.

1. Introduction

Private (on-premises) clouds have become a strategic alternative for organizations that require control over data, predictable performance, and compliance with regulatory or institutional constraints. Universities, research centers, and medium-sized enterprises increasingly adopt private clouds to host critical services, research platforms, and internal applications. While this model offers advantages in terms of governance and autonomy, it also introduces significant challenges for security monitoring, as services are distributed across virtual machines, containers, and heterogeneous networked components.

In such environments, the attack surface is inherently expanded. Virtualization layers, container platforms, and distributed services create multiple points of exposure, making incident detection and response more complex. In practice, effective security monitoring in private clouds depends on the ability to continuously collect, transport, and correlate logs from multiple sources in near real time. When adversarial activity escalates, the log ingestion and correlation pipeline itself may become a bottleneck, delaying detection and reducing the effectiveness of the Security Operations Center (SOC).

Security Information and Event Management (SIEM) systems play a central role in consolidating event logs, enriching contextual information, and correlating indicators of compromise to trigger alerts and support response activities. Although SIEM platforms are widely adopted in corporate and governmental environments, two relevant gaps remain in private-cloud scenarios. First, there is a lack of reproducible deployment models tailored to resource-constrained on-premises infrastructures, where elasticity is limited and operational simplicity is essential. Second, many experimental evaluations rely on offline datasets, synthetic traces, or post-mortem analysis, which provide limited insight into the real-time behavior of SIEM pipelines under bursty and adversarial workloads.

As a consequence, there is limited empirical understanding of how SIEM ingestion, parsing, and correlation stages behave under realistic attack dynamics and constrained computational resources. In particular, the interaction between attack intensity, workload burstiness, and server resource provisioning remains underexplored in the context of private clouds. This gap is critical, as under-provisioned SIEM deployments may exhibit delayed alerting, dropped events, or reduced visibility precisely when monitoring is most needed.

This paper addresses these gaps by implementing and evaluating a SIEM-as-a-Service model for private clouds using the open-source Wazuh platform. The solution is deployed using Docker Compose and integrated with Prometheus and Grafana to provide operational observability of the ingestion and correlation pipeline. To generate realistic adversarial conditions, we execute distributed SSH brute-force attacks using Hydra, modeling inter-arrival times with an exponential process to emulate burstiness and irregularity commonly observed in real attack traffic. We then evaluate how different attack intensities and server resource allocations affect alert generation and system behavior.

By combining a reproducible deployment model, controlled adversarial workload generation, and infrastructure-level observability, this work provides a systematic methodology to analyze non-linear effects and bottlenecks in SIEM pipelines deployed in private cloud environments. The objective is not to propose a new detection algorithm, but to offer empirical evidence and practical insights into the operational behavior of SIEM systems under adversarial load, supporting informed provisioning and deployment decisions.

Contributions. The main contributions of this work are:

- **A reproducible SIEM-as-a-Service deployment for private clouds**, based on Wazuh and Docker Compose, integrating a monitoring stack (Prometheus/Grafana/cAdvisor) to observe the SIEM pipeline under load.
- **A controlled adversarial workload for SSH brute-force generation**, using concurrent attackers and bursty inter-arrival times modeled by an exponential process, enabling repeatable stress conditions closer to real attack dynamics.
- **A multifactor experimental evaluation** that varies attack intensity and server resource allocation, producing a practical sizing baseline and analyzing non-linear effects and bottlenecks in the ingestion and correlation pipeline.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 details the proposed methodology and experimental design. Section 4 presents results and analysis. Finally, Section 5 concludes the paper and outlines future directions.

2. Related Work

Recent literature has extensively explored security monitoring, SIEM deployments, and threat detection mechanisms in cloud environments, with particular emphasis on scalability, automation, and integration with modern DevOps workflows. However, when focusing specifically on *private cloud* scenarios and on the *real-time operational behavior* of SIEM pipelines under adversarial load, important gaps remain.

A first group of studies addresses cloud security threats and mitigation strategies at a high level, often recommending layered defenses, continuous monitoring, and

governance frameworks [Ahmadi 2024, Kumar et al. 2023]. While these works provide valuable taxonomies and conceptual guidance, they typically do not deploy an operational SIEM pipeline under controlled stress conditions, nor do they analyze ingestion or correlation bottlenecks in resource-constrained private infrastructures. Similarly, comparative analyses of security frameworks, such as NIST, ISO, and CSA, help contextualize compliance requirements but offer limited empirical evidence regarding detection latency, queue buildup, or alert generation behavior under adversarial pressure [Chauhan and Shiaeles 2023].

A second group of works focuses on SIEM-as-a-Service and cloud-native SIEM architectures, aiming to improve elasticity, multi-tenancy, and deployment flexibility. Lopez et al. [López Velásquez et al. 2023], for example, discuss scalable SIEM models, but their evaluation does not combine bursty adversarial workloads with end-to-end observability of the ingestion and correlation pipeline. In parallel, studies that deploy Wazuh in cloud settings often emphasize feasibility, usability, or integration aspects [Amami et al. 2024, Moiz et al. 2024], yet provide limited systematic evaluation of operational behavior under controlled and progressively increasing attack intensity. As a result, the interaction between workload burstiness and resource provisioning remains largely unexplored in these contexts.

A third line of research investigates brute-force SSH detection using probabilistic, statistical, or machine-learning-based approaches. Singh et al. [Singh et al. 2023] and Rabbani et al. [Rabbani et al. 2024], for instance, propose detection models evaluated on offline datasets or simulated traces. While these contributions advance detection techniques, they typically do not assess how a real-time SIEM pipeline behaves when processing such attacks under constrained computational resources. Consequently, there is limited insight into ingestion delays, alert-generation instability, or non-linear performance degradation in operational deployments.

Other studies analyze throughput and performance aspects of monitoring systems in isolation. Sheeraz et al. [Sheeraz et al. 2024] investigate throughput under high event rates, but without coupling these measurements to realistic adversarial workloads or to a full SIEM correlation pipeline. Similarly, Yaker et al. [Yaker et al. 2024] discuss lightweight and adaptable monitoring solutions for constrained hosts, yet do not explore their behavior under bursty and escalating attack patterns.

In contrast to the above, this work jointly evaluates, in a real-time and reproducible manner: (i) a Wazuh-based SIEM-as-a-Service deployment tailored to private cloud environments; (ii) a controlled SSH brute-force workload with bursty inter-arrival times; and (iii) infrastructure-level observability using Prometheus and Grafana to analyze non-linear effects and bottlenecks in the SIEM ingestion and correlation pipeline. By explicitly combining adversarial workload generation with operational metrics, this study provides empirical evidence on how resource provisioning and attack intensity interact in private cloud deployments. Specifically, prior studies often address only a subset of the dimensions required for operational validation in private clouds—e.g., deployment feasibility without controlled adversarial workloads, or detection models without an end-to-end SIEM pipeline and infrastructure metrics. In contrast, our study explicitly combines real SSH attacks, containerized SIEM deployment, and operational metrics in a private-cloud setting, as summarized in Table 1.

Table 1. Comparison between representative related work and this study.

Work	Cld	OnP	Wz	SSH	Ctr	Met.
Adabi'23 [Adabi Raihan et al. 2023]	✓	–	–	–	–	CPU/RAM
Tuyishime'23 [Tuyishime et al. 2023]	✓	–	–	–	✓	lim.
Singh'23 [Singh et al. 2023]	–	–	–	✓	–	–
Yaker'24 [Yaker et al. 2024]	✓	✓	✓	–	–	–
Amami'24 [Amami et al. 2024]	✓	–	✓	–	✓	–
Moiz'24 [Moiz et al. 2024]	✓	–	✓	✓	–	–
Sheeraz'24 [Sheeraz et al. 2024]	–	–	–	–	–	thrpt
This	✓	✓	✓	✓	✓	CPU+alr

Legend: Cld=Cloud/SIEM; OnP=On-premise; Wz=Wazuh; SSH=Real SSH attacks; Ctr=Containerized deployment; Met.=operational metrics (CPU + alert volume); lim.=limited; thrpt=throughput; alr=alerts.

3. Methodology and Experimental Design

This study follows an experimental methodology structured around three main components: (i) a distributed system model for log collection and correlation using a SIEM-as-a-Service deployment; (ii) a controlled adversarial workload based on SSH brute-force attacks with bursty inter-arrival times; and (iii) a multifactor experimental design that varies attack intensity and server resource allocation. The objective is to analyze how the SIEM ingestion and correlation pipeline behaves under realistic adversarial load in a private cloud environment.

3.1. System Model and SIEM Pipeline

The experimental environment is composed of two virtual machines (VMs) connected through a private network, deployed on a KVM-based private cloud infrastructure, emulating a monitored endpoint and a centralized SIEM service. This design reflects a common deployment pattern in private clouds, where application workloads and security management services are logically separated. Such a topology is representative of on-premises environments in universities, research centers, and medium-sized organizations, in which security management services are centralized while application workloads remain distributed.

Table 2 summarizes the physical infrastructure, virtualization environment, network access model, and deployment topology used in the experiments.

All experiments were conducted on real virtual machines and containers running on physical hardware, without the use of network simulators, trace replays, or synthetic log generators. This ensures that all observed effects reflect the behavior of a real SIEM pipeline under adversarial load.

VM1 – SIEM and Monitoring Server. VM1 hosts the Wazuh SIEM stack, including the manager, indexer, API, and dashboard components. It is also responsible for running the observability stack composed of Prometheus, Grafana, and cAdvisor. This

Table 2. Experimental infrastructure and deployment environment.

Component	Description
Physical host	Ubuntu 24.04 LTS server, Intel Core I7 Processor (Haswell architecture, 3.6 GHz, 16 MB cache, 32GB RAM), member of a private cloud with 45 active physical nodes at LaSDPC (ICMC–USP).
Virtualization layer	KVM-based virtualization environment hosting all experimental virtual machines.
Virtual machines	Two VMs running Ubuntu 24.04 LTS, with CPU and memory resources dynamically adjusted according to experimental scenarios.
Container platform	Docker used inside each VM to deploy Wazuh stack, monitoring services, and attack generation components.
Network access	Managed by a ClearOS gateway with SSH port forwarding for remote access; access is also available via VPN.
Domain and gateway	Private cloud accessible via `andromeda.lasdpc.icmc.usp.br` (ClearOS gateway).
Experimental topology	VM1: Wazuh SIEM and monitoring stack; VM2: monitored SSH service and Kali Linux container for attack generation.
Reproducibility resources	All scripts, Docker Compose files, and configurations publicly available at: `https://github.com/ICMC-SSC5973-2025/projeto-ssc5973-grupo-01`.

VM receives security events from monitored hosts, performs rule evaluation and correlation, and generates alerts that are visualized in the dashboard.

VM2 – Monitored Endpoint and Attack Source. VM2 hosts the monitored SSH service and the Wazuh agent. In addition, it runs a Kali Linux container used to generate brute-force attacks with Hydra. Although the attacker and the target reside on the same VM in this experimental setup, they are isolated in separate containers, which preserves network semantics while enabling controlled and reproducible workload generation. While this setup does not aim to model wide-area latency, it preserves the end-to-end SIEM pipeline (log generation, agent collection, transport, parsing, correlation, and alerting) under controlled and repeatable adversarial load.

Figure 1 summarizes the experimental environment and the placement of the SIEM and monitoring components. The logical pipeline of events is as follows: (1) SSH authentication attempts are generated against the target service; (2) system logs are collected by the Wazuh agent; (3) events are forwarded to the Wazuh manager on VM1; (4) the SIEM performs parsing, rule matching, and correlation; and (5) alerts are generated and visualized in the dashboard. In parallel, Prometheus continuously scrapes operational metrics from the host and containers, enabling the analysis of resource utilization and system behavior during the experiments.

This architecture enables the systematic observation of non-linear effects in the ingestion and correlation pipeline, such as queue buildup, processing delays, and resource contention, which are critical in distributed monitoring scenarios.

Based on this deployment model, the experimental workload was designed to exercise the full SIEM pipeline, from log generation at the endpoint to correlation and alerting at the centralized manager, under progressively increasing adversarial pressure.

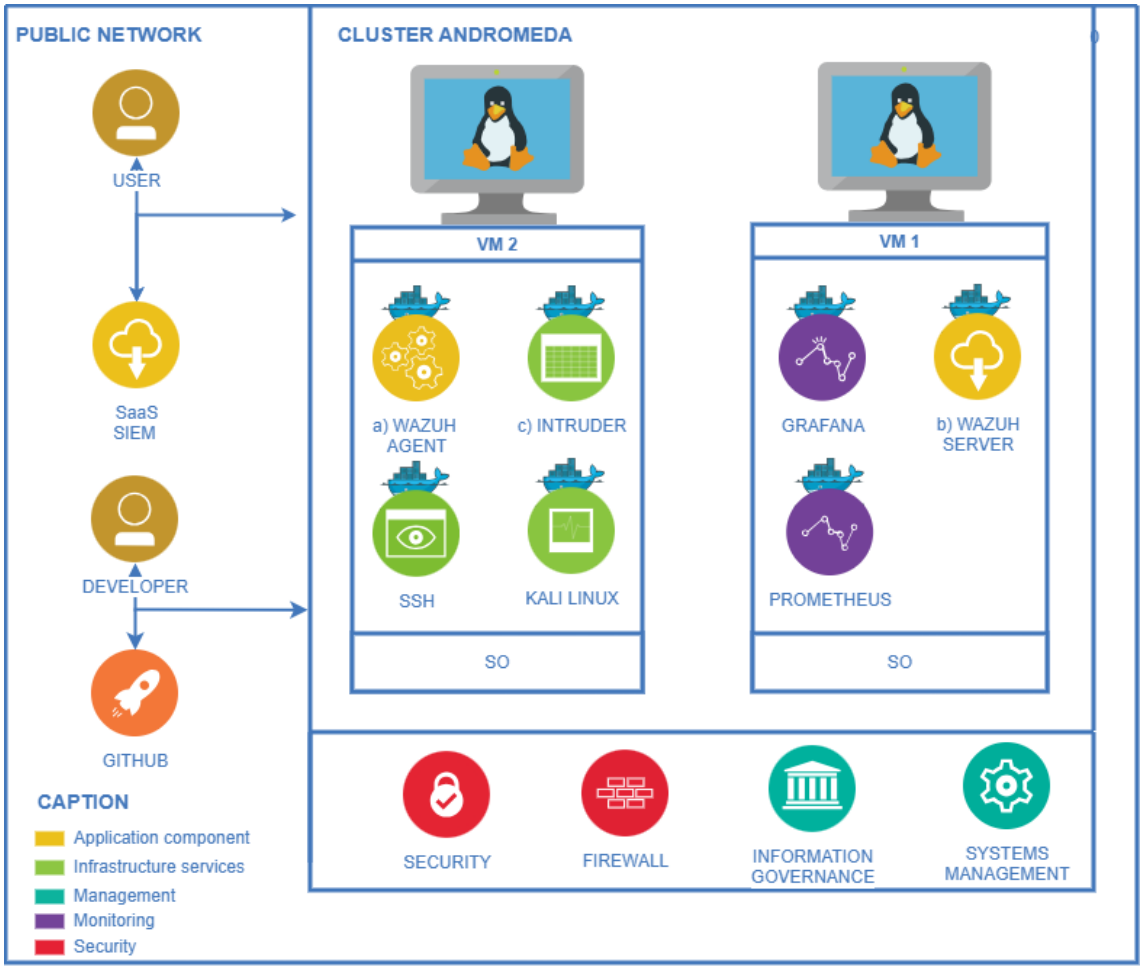

Figure 1. Experimental architecture with two virtual machines (VM1: SIEM/monitoring; VM2: monitored endpoint and attacker container) in a private-cloud setting.

3.2. Adversarial Workload: SSH Brute-Force Attacks

To generate realistic security events, we perform SSH brute-force attacks using Hydra [Hydra 2023]. Attacks are launched from the Kali Linux container against the SSH service running on the monitored environment. Synthetic username and password lists are used to ensure reproducibility across experimental runs.

Unlike constant-rate traffic generation, the inter-arrival times between attack attempts are modeled using an exponential distribution. This choice provides a simple and reproducible way to introduce irregularity and burstiness in the attack stream, while keeping the workload generation controllable. We note that real-world adversarial traffic may exhibit more complex temporal patterns; nevertheless, the proposed model is sufficient to systematically stress the SIEM ingestion and correlation pipeline under progressively increasing adversarial pressure.

The exponential model is a standard baseline to represent inter-arrival times of independent events and is widely used in network-traffic modeling. Although real traffic may present more complex burstiness, prior work indicates that exponential inter-arrival modeling remains useful to capture essential arrival-time properties and to represent irregular temporality in cyberattack processes [Paxson and Floyd 1995, Liu et al. 2019].

Let T denote the inter-arrival time between successive attack attempts, defined as:

$$T = -\frac{\ln(1-U)}{\lambda}, \quad U \sim Uniform(0,1)$$

where λ is the rate parameter. To simulate progressive escalation of attack intensity, λ is updated after each cycle according to:

$$\lambda_{n+1} = \lambda_n \cdot e^k$$

where k is a growth coefficient.

This mechanism produces a workload with increasing frequency of attempts and burstiness over time, stressing the SIEM pipeline in a controlled manner. This workload design allows us to evaluate not only detection capability but also the dynamic behavior of the ingestion and correlation stages under fluctuating and escalating adversarial pressure.

3.3. Experimental Factors and Levels

The experiments follow a multifactor design with two independent variables: attack intensity and server resource allocation. In this design, the exponential model controls the *temporal irregularity* between attack cycles, whereas Factor F1 controls the *concurrency level* of the brute-force workload (number of attackers and Hydra threads). Table 3 summarizes the factors and levels considered in the experimental design.

Table 3. Experimental factors and levels used in the SSH brute-force benchmark.

Factor	Level 1 (Low)	Level 2 (Medium)	Level 3 (High)
F1 – Attack intensity (SSH brute-force workload)			
Number of attackers (terminals)	1	3	5
Hydra threads per attacker	4	16	32
F2 – SIEM server resource allocation (VM1)			
vCPU	4	6	8
Memory (GB RAM)	8	12	16

Attack intensity (F1). Attack intensity is controlled by varying the number of concurrent attackers and the number of threads used by Hydra. Three levels are considered:

- Low: 1 terminal, 4 threads;
- Medium: 3 terminals, 16 threads;
- High: 5 terminals, 32 threads.

Server resource allocation (F2). The SIEM server (VM1) is configured with three different resource profiles:

- 4 vCPUs and 8 GB of RAM;
- 6 vCPUs and 12 GB of RAM;
- 8 vCPUs and 16 GB of RAM.

Each combination of F1 and F2 is executed independently, resulting in nine experimental scenarios. This full-factorial design enables the analysis of both main effects and interaction effects between workload intensity and resource provisioning. Between consecutive runs, a waiting interval is enforced to avoid interference and residual effects.

3.4. Monitoring and Metrics

Two classes of metrics are collected during the experiments.

Detection metrics. From the Wazuh dashboard and alert logs, we extract:

- number of detected SSH brute-force events;
- alert generation behavior under each workload;
- qualitative observation of false positives or missed detections, when present.

Operational metrics. From Prometheus and Grafana, we collect:

- CPU usage of the SIEM server (peak and average) during the attack windows;
- container-level resource utilization for sanity-checking the deployment behavior.

These metrics enable the analysis of how increasing adversarial load and different resource configurations affect the performance and stability of the SIEM pipeline.

3.5. Execution Procedure

For each experimental scenario, the following procedure is applied:

1. configure VM1 with the target resource profile;
2. start the Wazuh and monitoring services;
3. launch the SSH brute-force workload with the defined intensity level;
4. monitor alert generation and resource utilization during the attack period;
5. collect and store metrics for subsequent analysis.

Experiments conducted. A total of nine experiments were conducted, combining the factor levels in Table 3. For each VM1 resource profile (4 vCPUs/8 GB, 6 vCPUs/12 GB, and 8 vCPUs/16 GB), we executed the low, medium, and high attack-intensity workloads on VM2. To reduce interference between consecutive runs, we enforced a cooldown interval between experiments and performed runs on different days when needed. Detection statistics (peak and average password-guessing events) were obtained from the Wazuh Dashboard, while peak and average CPU utilization were collected from Grafana during the attack windows. This controlled procedure ensures repeatability and allows fair comparison across all factor combinations.

4. Results and Analysis

This section presents and analyzes the experimental results obtained from the nine scenarios defined by the combination of attack intensity (F1) and server resource allocation (F2). The analysis focuses on two complementary dimensions: (i) detection behavior and alert generation, and (ii) operational performance of the SIEM ingestion and correlation pipeline under increasing adversarial load.

4.1. Alert Generation and Detection Behavior

Figure 2 summarizes the peak and average number of detected password-guessing (SSH brute-force) events across all experimental scenarios.

As expected, alert volume increases with attack intensity across all resource profiles, reflecting the higher number of authentication attempts generated by the workload.

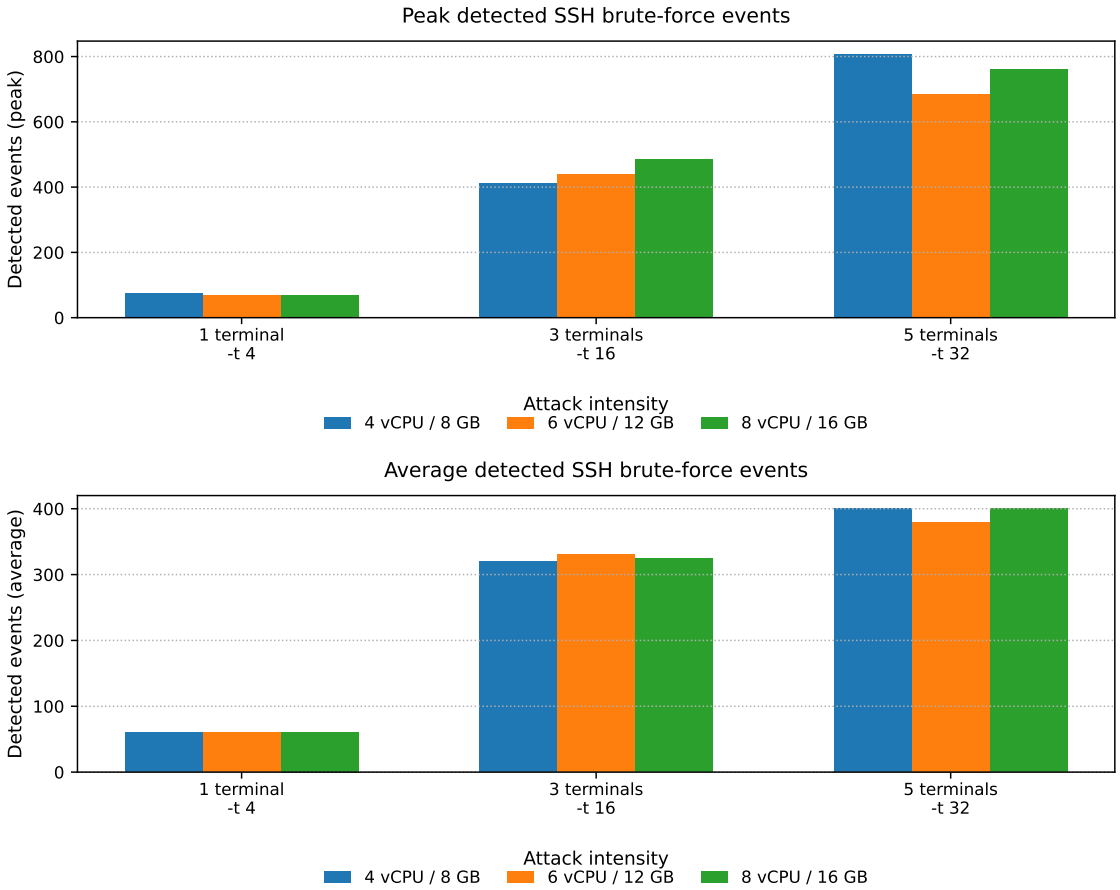

Figure 2. Detected SSH brute-force (password-guessing) events under different attack intensities and server configurations: peak (top) and average (bottom).

However, this growth is not strictly linear and threshold-driven behavior becomes evident as the SIEM server operates under constrained resources. In the configuration with 4 vCPUs and 8 GB RAM, alert generation becomes unstable under high attack intensity, with noticeable fluctuations and delayed correlation. This behavior indicates saturation in the SIEM ingestion and processing pipeline, where events accumulate faster than they can be parsed, correlated, and converted into alerts. In practical terms, this corresponds to reduced situational awareness precisely when adversarial pressure is highest.

In contrast, the configuration with 8 vCPUs and 16 GB RAM maintains a more regular and stable alerting pattern, even under high attack intensity. This suggests that additional computational resources effectively mitigate queue buildup and processing delays, allowing the pipeline to absorb bursts of events without losing consistency. An important observation is that increasing attack intensity does not always translate proportionally into alert volume. In some scenarios, particularly under constrained resources, the SIEM generates fewer alerts than expected. Rather than indicating improved security, this behavior is symptomatic of processing bottlenecks, delayed correlation, or potential event drops. From an operational perspective, this represents a critical risk: reduced alert volume may mask ongoing attacks instead of signaling their absence.

4.2. CPU Utilization

Figure 3 presents the peak and average CPU utilization of the SIEM server during the experiments.

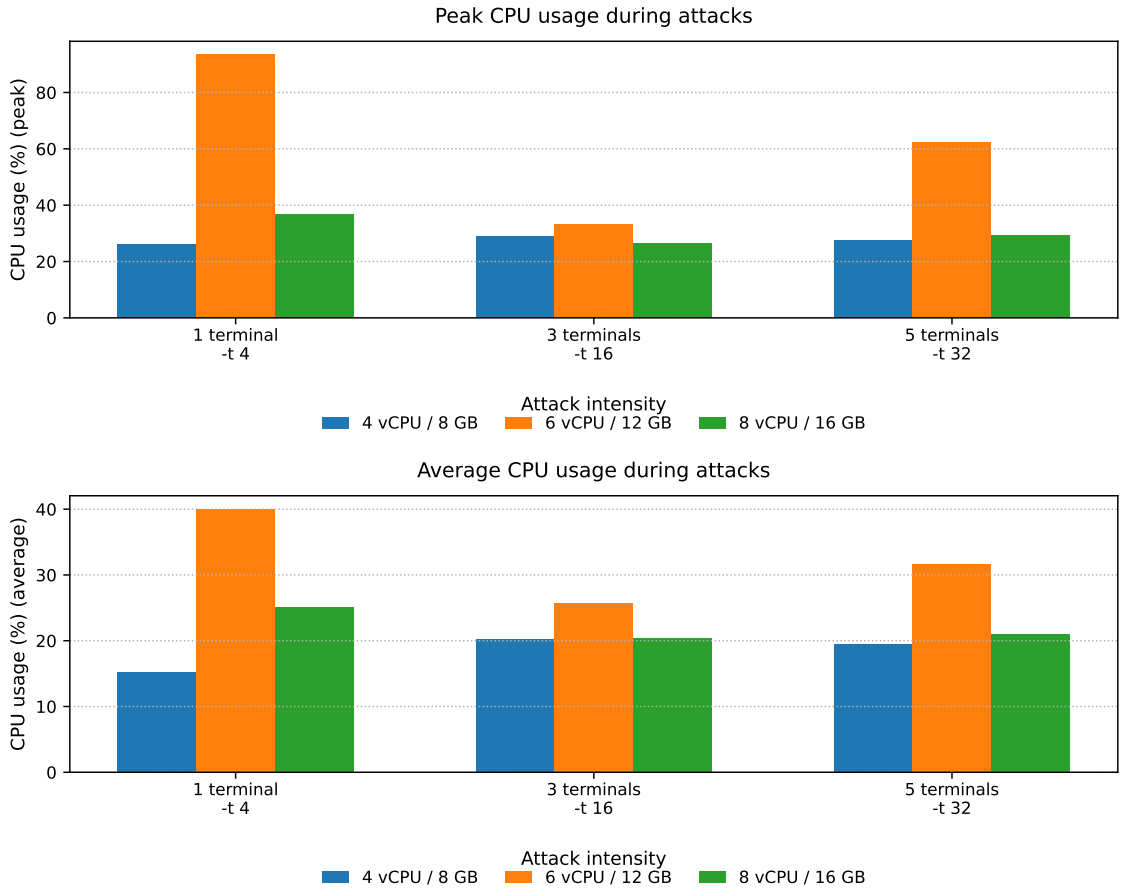

Figure 3. CPU utilization of the SIEM server during attacks for each experimental scenario: peak (top) and average (bottom).

Under low and medium attack intensities, all configurations exhibit stable and predictable CPU utilization, indicating that the SIEM pipeline operates within comfortable capacity limits. However, under high attack intensity, clear differences emerge between resource profiles.

The configuration with 4 vCPUs and 8 GB RAM reaches sustained high CPU utilization, often approaching saturation. This sustained pressure corroborates the alert-generation instability observed in Figure 2, reinforcing the interpretation that the ingestion and correlation stages are operating near their processing limits.

The 6 vCPUs and 12 GB RAM configuration shows improved stability compared to the lowest profile, but still presents peaks and variability under high adversarial load. In contrast, the 8 vCPUs and 16 GB RAM configuration maintains more regular CPU usage and lower variance, indicating a greater capacity to absorb bursty event streams without triggering saturation effects.

Although memory usage and internal buffering mechanisms also influence pipeline behavior, CPU utilization and alert-generation stability emerge as primary in-

dicators of operational robustness in the evaluated scenarios.

4.3. Interaction Between Workload and Resource Provisioning

The joint analysis of detection behavior and CPU utilization reveals a clear interaction effect between attack intensity (F1) and server resource provisioning (F2). While low-intensity attacks are handled adequately by all configurations, medium and high-intensity workloads progressively expose the limitations of under-provisioned environments.

In particular, the transition from medium to high attack intensity produces a disproportionate impact on the 4 vCPUs and 8 GB RAM configuration, whereas the impact on the 8 vCPUs and 16 GB RAM configuration is more gradual. This non-linear behavior confirms that SIEM pipelines are subject to threshold effects, where performance degrades rapidly beyond certain load levels.

From a systems perspective, this behavior is consistent with queueing dynamics in multi-stage processing pipelines: once arrival rates approach or exceed service rates, small increases in workload lead to large increases in latency, queue length, and instability. In security operations, this translates directly into delayed or missing alerts during peak attack periods. These results support the claim that resource provisioning should be guided not only by average operational load, but also by worst-case adversarial scenarios. Underestimating attack intensity may lead to blind spots in monitoring precisely when visibility is most critical.

4.4. Discussion

From a practical perspective, the results indicate that the configuration with 8 vCPUs and 16 GB RAM provides the most stable behavior across all tested scenarios, combining high alerting capacity with regular and predictable resource utilization. This configuration consistently absorbs bursty adversarial traffic without exhibiting saturation effects, pronounced instability, or loss of alert consistency.

In contrast, the 4 vCPUs and 8 GB RAM configuration, although potentially attractive from a cost-efficiency standpoint, shows clear limitations under moderate and high attack intensities. In these scenarios, the SIEM pipeline becomes unstable, with visible fluctuations in alert generation and sustained high CPU utilization.

From an operational viewpoint, this behavior is particularly critical, as it indicates reduced observability precisely when adversarial pressure is highest. The 6 vCPUs and 12 GB RAM configuration represents an intermediate case, offering improved stability compared to the lowest profile, but still exhibiting variability under high load. This suggests that incremental resource increases may alleviate, but not fully eliminate, non-linear effects in the processing pipeline when attack intensity grows. These observations highlight a fundamental characteristic of SIEM systems deployed in private clouds: performance degradation under adversarial load is not gradual, but threshold-driven. Once the ingestion and correlation stages approach their processing limits, small increases in event arrival rate can lead to disproportionate impacts on latency, queue buildup, and alert stability. This behavior is consistent with queueing dynamics in multi-stage distributed pipelines and reinforces the importance of evaluating systems under peak conditions rather than average workload.

From a systems-engineering perspective, the results emphasize that under-provisioning a SIEM infrastructure does not merely reduce performance, but can actively compromise detection capability and situational awareness. In private cloud environments, where elasticity is inherently limited compared to public cloud platforms, this risk is amplified. Consequently, sizing decisions must explicitly account for worst-case adversarial scenarios rather than relying solely on nominal operational load. It is also worth noting that the observed behavior is not specific to the Wazuh platform, but rather reflects general properties of log ingestion and correlation pipelines under bursty workloads. As such, the insights derived from this study are applicable to a broader class of SIEM and security-monitoring systems deployed in on-premises environments.

Finally, while the experimental workload focuses on SSH brute-force attacks, the methodology is intentionally generic. The same experimental design can be applied to other attack classes, protocols, and services, enabling systematic evaluation of SIEM behavior under diverse adversarial conditions.

5. Conclusion

This paper presented a reproducible SIEM-as-a-Service deployment for private clouds based on Wazuh, containerized with Docker Compose and instrumented with Prometheus and Grafana for operational observability. Unlike studies that rely on offline datasets or synthetic traces, the proposed approach evaluates the SIEM ingestion and correlation pipeline under controlled and realistic adversarial workloads, using distributed SSH brute-force attacks with bursty inter-arrival times. Through a multifactor experimental design, varying both attack intensity and server resource allocation, we demonstrated that SIEM behavior under adversarial load is strongly non-linear and highly sensitive to resource provisioning. The results show that configurations with limited resources may exhibit instability, delayed correlation, and reduced alert visibility under high attack intensity, whereas more adequately provisioned configurations maintain stable and predictable behavior. In particular, the configuration with 8 vCPUs and 16 GB RAM consistently provided the most robust performance across all tested scenarios, combining high alerting capacity with regular resource utilization. These findings reinforce the importance of sizing SIEM infrastructure based on worst-case adversarial conditions rather than average operational load, especially in private cloud environments where dynamic elasticity is limited.

Beyond the specific experimental results, the main contribution of this work lies in the methodology itself. By integrating controlled adversarial workload generation, multifactor experimental design, and infrastructure-level observability, we provide a systematic and reproducible approach to evaluate SIEM behavior under realistic conditions. This methodology can be readily adapted to other infrastructures, deployment models, and attack scenarios, supporting informed decision-making in security operations and infrastructure planning.

As future work, we plan to extend the experimental framework to additional attack classes (e.g., lateral movement, data exfiltration, and denial-of-service), evaluate memory and I/O bottlenecks in greater depth, and explore distributed and multi-tenant SIEM deployments. We also intend to investigate automated scaling strategies and adaptive resource management policies for SIEM services in private cloud environments. Overall, this study contributes empirical evidence and methodological guidance for the design, siz-

ing, and evaluation of SIEM systems in on-premises distributed infrastructures, bridging a gap between conceptual security architectures and their real-world operational behavior under adversarial pressure.

References

Adabi Raihan, M., Sukarno, P., and Wardana, A. A. (2023). Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning. *Procedia Computer Science*, 217:1406–1415.

Ahmadi, S. (2024). Systematic literature review on cloud computing security: Threats and mitigation strategies. *Journal of Information Security*, 15:148–167.

Amami, R., Charfeddine, M., and Masmoudi, S. (2024). Exploration of open source siem tools and deployment of an appropriate wazuh-based solution for strengthening cyberdefense. In *10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1–7.

Chauhan, M. and Shiaeles, S. (2023). An analysis of cloud security frameworks, problems and proposed solutions. *Network*, 3(3):422–450.

Hydra (2023). Hydra tool documentation - vr. 9.6 - 2023. Available at: `https://www.kali.org/tools/hydra/#tool-documentation`. Accessed: Aug. 28, 2025.

Kumar, S., Rajlingam, A., and Gokila, B. (2023). Study analysis of cloud security challenges and issues in cloud computing technologies. *Journal of Science, Computing and Engineering Research*, 6(8):6–11.

Liu, Y., Da Silva, I., and Joshi, R. (2019). Modeling adversarial network traffic with stochastic arrival processes. *Computer Networks*.

López Velásquez, J., Martínez Monterrubio, S., Sánchez Crespo, L., et al. (2023). Systematic review of siem technology: Siem-sc birth. *International Journal of Information Security*, 22:691–711.

Moiz, S., Majid, A., Basit, A., Ebrahim, M., Abro, A. A., and Naeem, M. (2024). Security and threat detection through cloud-based wazuh deployment. In *IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, pages 1–5.

Paxson, V. and Floyd, S. (1995). Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*.

Rabbani, M., Gui, J., Nejati, F., Zhou, Z., Kaniyamattam, A., Mirani, M., Piya, G., Opushnyev, I., Lu, R., and Ghorbani, A. (2024). Device identification and anomaly detection in iot environments. *IEEE Internet of Things Journal*, 12(10):13625–13643.

Sheeraz, M., Durad, M. H., Paracha, M. A., Mohsin, S. M., Kazmi, S. N., and Maple, C. (2024). Revolutionizing siem security: An innovative correlation engine design for multi-layered attack detection. *Sensors*, 24(15):4901.

Singh, V. et al. (2023). Brutector: A probabilistic detection model for bruteforce attacks in ssh server. In *16th International Conference on Security of Information and Networks (SIN)*, pages 1–8.

Tuyishime, E., Balan, T. C., Cotfas, P. A., Cotfas, D. T., and Rekeraho, A. (2023). Enhancing cloud security—proactive threat monitoring and detection using a siem-based approach. *Applied Sciences*, 13(22):12359.

Yaker, K., Ait Salem, B., Pierard, B., AitSaadi, N., and Raynal, V. (2024). A novel edge siem for industrial iot flows within 5g private networks. In *Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6.