RESEARCH-ARTICLE

# Zero Trust Architecture for Ransomware Defense in Virtualized Environment

**ATHARVA DHUMAL**, National College of Ireland, Dublin, Ireland

**MUSTAFA GHALEB**, Kocaeli University, Izmir, Kocaeli, Turkey

**SAMAH ABDELSALAM**, University of Ha'il, Ha'il, Ha'il, Saudi Arabia

**ARGHIR NICOLAE MOLDOVAN**, National College of Ireland, Dublin, Ireland

**MOSAB HAMDAN**, National College of Ireland, Dublin, Ireland

**Open Access Support** provided by:

**University of Ha'il**

**National College of Ireland**

**Kocaeli University**



**Published:** 01 December 2025

**Citation in BibTeX format**

BDCAT '25: IEEE/ACM 12th International Conference on Big Data Computing, Applications and Technologies
*December 1 - 4, 2025*
*Nantes, France*

**Conference Sponsors:**
SIGARCH

# Zero Trust Architecture for Ransomware Defense in Virtualized Environment

Atharva Dhumal
School of Computing
National College of Ireland
Dublin, Ireland
x23404388@student.ncirl.ie

Mustafa Ghaleb
Department of Software Engineering
Kocaeli University
Kocaeli, Turkiye
mustafa.ghaleb@kocaeli.edu.tr

Samah Abdelsalam
Department of Management
Information System
University of Hail
Hail, Saudi Arabia
samah.gubar@uoh.edu.sa

Arghir-Nicolae Moldovan
School of Computing
National College of Ireland
Dublin, Ireland
arghir.moldovan@ncirl.ie

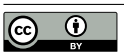
Mosab Hamdan
School of Computing
National College of Ireland
Dublin, Ireland
mosab.mohamed@ncirl.ie

## Abstract

The ongoing surge of ransomware has underscored the need to shift from perimeter-based security to Zero Trust models. This paper investigates a Zero Trust Architecture (ZTA) approach to containing ransomware in a virtualized environment using least-privilege controls, micro-segmentation, and continuous monitoring. We develop an open-source, lightweight security architecture comprising Wazuh for real-time auditing and alerts, audited for system logging, and the Uncomplicated Firewall (UFW) for network segmentation within a VirtualBox laboratory network, consisting of Ubuntu as the victim and Kali as the attacker virtual machines. A simulated ransomware attack is conducted to evaluate detection latency, data impact, system overhead, and alert accuracy. The prototype ZTA framework detected ransomware activity in an average of $\approx$ 5.3 seconds. This detection limited encryption to approximately 20% of files prior to the activation of containment measures, while maintaining minimal CPU and memory overhead and exhibiting a low rate of false positives. These findings illustrate the successful early containment of ransomware via the implementation of Zero Trust controls. Although evaluated in a laboratory environment, the methodology is applicable to trustworthy and secure cloud or hybrid systems by improving data protection, facilitating compliance-oriented audits, and minimizing the impact of attacks.

## CCS Concepts

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; *Access control*; *Virtualization and security*; • **Computer systems organization** → *Cloud computing*.




*BDCAT '25, Nantes, France*

ACM ISBN 979-8-4007-2286-8/25/12
https://doi.org/10.1145/3773276.3774876

## Keywords

Zero Trust Architecture, Ransomware, Micro-Segmentation, Wazuh, Auditd, Uncomplicated Firewall, Virtualized Lab, Ransomware Containment

**ACM Reference Format:**
Atharva Dhumal, Mustafa Ghaleb, Samah Abdelsalam, Arghir-Nicolae Moldovan, and Mosab Hamdan. 2025. Zero Trust Architecture for Ransomware Defense in Virtualized Environment. In *IEEE/ACM 12th International Conference on Big Data Computing, Applications and Technologies (BDCAT '25), December 01–04, 2025, Nantes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3773276.3774876

## 1 Introduction

Ransomware continues to be one of the most destructive cyber threats, causing significant financial and operational damage across various industries. According to the State of Ransomware 2024 report by Sophos, 59% of organizations experienced ransomware attacks [3] in the preceding year. Among those who opted to pay the ransom, the average amount disbursed was $12.0 million. Furthermore, the report indicates a deceleration in recovery efforts, with only 35% achieving full recovery within one week, while 34% required more than a month to do so[1]. These statistics underscore the insufficiency of traditional perimeter-based defense mechanisms. This urgency has fostered an increased interest in Zero Trust Architecture (ZTA) as a more robust defensive framework. ZTA operates on the principle of "never trust, always verify," necessitating ongoing identity verification, stringent access controls, and continuous monitoring of every user and device. Trust has been defined and classified by many researchers [1, 9]. This approach avoids any assumption of implicit trust within the perimeter of a network [10]. This paradigm effectively addresses the deficiencies inherent in traditional models by implementing rigorous identity verification, detailed access control, and continuous behavioral monitoring.

The momentum for the adoption of ZTA has notably increased in recent years. Significantly, the 2021 Zero Trust Maturity Model issued by the U.S. Cybersecurity and Infrastructure Security Agency

[1] https://newsletter.radensa.ru/wp-content/uploads/2024/05/sophos-state-of-ransomware-2024-wp.pdf

(CISA)[2] has urged organizations to expedite ZTA implementation. Our work contributes to this movement by building a practical ZTA prototype in a virtualized lab environment and demonstrating its effectiveness against ransomware. The proposed solution utilizes exclusively open-source tools, specifically Wazuh for real-time monitoring and alerting, Linux auditd for system auditing, and the Uncomplicated Firewall (UFW) for host-level micro-segmentation. It enforces identity verification primarily at the host level using least-privilege, non-interactive service accounts and OS permissions. Centralized identity services like multi-factor authentication (MFA) or single sign-on (SSO) are not integrated in this prototype. The framework is implemented on lightweight VirtualBox virtual machines, facilitating an end-to-end ransomware isolation workflow that is easily deployable in environments with limited resources or in educational settings that do not possess enterprise infrastructure. This design separates identity control, behavioral monitoring, and network segmentation, providing flexibility for various environments. Our ransomware simulations confirm the effectiveness of each layer. In addition to the implementation, we offer reusable artifacts, including custom auditd rules, firewall configurations, and a containment playbook, to facilitate broader adoption and to promote further scholarly inquiry into this methodology.

This study aims to achieve three main contributions. Firstly, we investigate whether the implementation of stringent identity verification mechanisms, such as MFA and service-level credentials, can prevent unauthorized access at the outset of a ransomware incident. Secondly, we evaluate the effectiveness of fine-grained micro-segmentation to hinder attackers from executing lateral movements to other systems following an initial breach. Thirdly, we assess the potential of continuous behavioral monitoring of system activities to swiftly detect ransomware-like conduct and automatically initiate alerts or defensive actions in response to the threat.

The remainder of this study is organized as follows: Section 2 provides a review of the literature concerning ZTA and strategies for containing ransomware. Section 3 examines the design and implementation of the proposed ZTA Ransomware Defense framework, focusing on tool configuration, attack simulations, and data collection methods. Section 4 presents and analyzes the results of the ransomware simulations, assessing detection latency, containment efficacy, resource utilization, and alert precision. Section 5 concludes with a summary of the findings, discusses the limitations of the study, and suggests directions for future research.

## 2 Related Work

Researchers have commenced the incorporation of Zero Trust principles into ransomware defense strategies; nonetheless, a considerable number of solutions are tailored to specific elements, consequently leaving notable gaps. Initial methodologies emphasize real-time surveillance and deceptive tactics: for example, Zhuravchak et al. [15] employ honeypots integrated with eBPF to observe attackers in real-time, thus enhancing detection and response mechanisms. Alternative approaches prioritize isolated analysis environments. As demonstrated by Companucci et al. [7], they propose an air-gapped ransomware analysis framework (SAFARI) that facilitates secure, reproducible experiments and automated countermeasures, whereas von der Assen et al. [14] demonstrate an interception of file I/O at the operating system level (GuardFS) to anticipate ransomware activity and mitigate potential damage. These initiatives highlight the progression towards a behavior-oriented, proactive defense mechanism that is congruent with Zero Trust principles.

Numerous studies incorporate open-source tools, such as Wazuh, in the enforcement of ZTAs. Chamkar et al. [6] enhance Wazuh's monitoring capabilities by integrating machine learning techniques, which elevates detection accuracy to over 95%, albeit with significant computational demands. In contrast, our method emphasizes a more lightweight implementation alongside practical containment metrics. We illustrate that an open-source stack is capable of detecting ransomware within $\sim$ 5 seconds and mitigating $\sim$ 80–88% of the file encryption impact. This is achieved while maintaining minimal (single-digit percentage) CPU overhead, a balance that is frequently overlooked in prior research.

The studies presented in Table 1 delineate three interrelated trajectories for Zero-Trust-aligned ransomware defense strategies: (i) behavior-focused visibility and alerting systems employing open-source stacks mapped to the MITRE ATT&CK framework, such as Wazuh agents and real-time triage rules [4, 5, 11]; (ii) segmentation-initiated containment mechanisms designed to restrict lateral movement, including micro-segmentation in enterprise or SDN environments [4, 12]; and (iii) data-driven analytic methodologies aimed at enhancing detection accuracy, including rule-based and machine learning pipelines, along with hybrid machine learning incorporating indicators of compromise (IOCs) [2, 6, 13]. Complementary initiatives underscore the importance of safe experimentation and early interception—such as air-gapped, reproducible analysis frameworks (SAFARI) and OS-level file I/O interceptions (GuardFS), to expedite response times and reduce operational losses [7, 14], while the use of eBPF honeypots enhances adversary observability [15]. Nevertheless, operational evidence on containment (e.g., percentage of encryption thwarted during live scenarios) and runtime cost profiling are less frequently reported than metrics related to accuracy or visibility. This research addresses this shortcoming by providing a fully reproducible, dual-VM, open-source testbed that assesses detection latency ($\approx$ 5.3 s), the effectiveness of containment strategies ($\approx$ 80–88% of files preserved), and overhead ($\approx$ an 8% CPU monitoring overhead with total CPU usage peaking at 65–72% during an attack), while implementing host-centered identity controls (such as non-interactive accounts and permission hardening) and micro-segmentation. This positions our prototype as a resource-efficient and easily adoptable reference solution that complements previous accuracy-focused studies by incorporating measured, end-to-end containment, while also being readily adaptable for securing virtualized, hybrid, or cloud infrastructures without the need to alter the core methodology.

## 3 Methodology: ZTA Ransomware Defense

In this work, we adopt a Design Science Research (DSR) approach to conduct a practical evaluation of ZTA for the containment of ransomware. This covers both the execution of ransomware attacks and their subsequent containment. The DSR methodology is particularly appropriate in this context, as it facilitates the design, implementation, and assessment of a functional proof-of-concept

[2] https://www.cisa.gov/resources-tools/resources/zero-trust-maturity-model

**Table 1: Summary of Related Work on Zero Trust and Ransomware Detection**

| Study / Author(s) | Focus Area | Key Tools/Techniques | Key Findings / Contributions |
|---|---|---|---|
| Berar [5] | Wazuh in Zero Trust enforcement | Wazuh agent-manager, MITRE ATT&CK | Emphasised real-time visibility, behavioral logging, and rule-based alerts |
| Roy et al. [11] | Threat mapping in ZTA | MITRE ATT&CK + Wazuh rules | Swift triage and alerting for adversarial patterns |
| Basta et al. [4] | Virtual domain protection using Wazuh | Micro-segmentation + Wazuh logging | Ensured lateral movement restriction and asset verification |
| Chamkar et al. [6] | Continuous monitoring with Wazuh | Rule-based + ML-enhanced detection | Achieved 97% accuracy in detecting ransomware behaviors |
| Gambo and Almulhem [8] | Low-cost ZTA implementation | UFW, iptables, Wazuh | Demonstrated affordable Zero Trust on VMs for sectors with limited budgets |
| Alzubi et al. [2] | AI for enhancing ransomware detection | Hybrid ML with IOCs | Achieved 94% F1-score on CIC IDS 2018 dataset |
| Svet et al. [13] | Clustering malicious behavior | Unsupervised clustering | Detected ransomware without labelled data |
| Sakhi [12] | Micro-segmentation in ZTA to reduce attack surfaces and limit lateral movement. | SDN, policy enforcement engines, cloud-native segmentation, traffic flow analysis, and testbed evaluation. | Reduces lateral movement, granular policies improve security, automation is essential, viable for SMEs, proposed adaptable ZTA framework. |

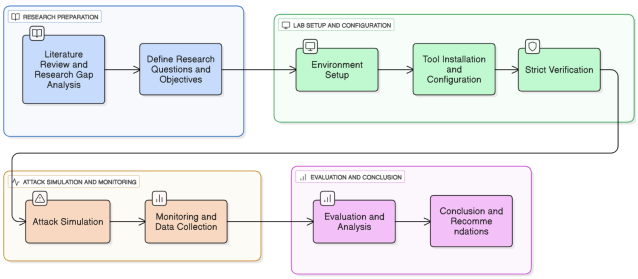



**Figure 1: Research Framework**

security solution. Our experimental framework, depicted in Figure 1, serves as a tool to elucidate the influence of each ZTA mechanism within an active ransomware scenario.

To provide a clear, step-by-step view of the implementation, we formalized the procedure as an experimental methodology, presented in Figure 2. This methodology encapsulates all key procedures from virtual machine setup and baseline security configuration, through monitoring deployment and ransomware simulation, to detection, response, and performance analysis. Two virtual machines were used: an Ubuntu 22.04 server as the victim endpoint and a Kali Linux as the attacker, both running on Oracle VirtualBox with a host-only network ("ZTA-Net") to emulate a segmented enterprise network. This isolated setup ensures no external interference and reflects ZTA's micro-segmentation ethos by containing attack traffic within a closed environment. This structured approach enhances reproducibility, allowing other researchers to replicate the experiment in their own testbeds. The proposed ZTA has been made accessible on GitHub to facilitate its reproducibility [3].

## 3.1 Security Stack & Configuration

We deploy a fully open-source stack on the Ubuntu victim VM: Wazuh (manager+agent) and Linux `auditd` for continuous monitoring with ATT&CK-mapped rules, and UFW for micro-segmentation. Wazuh correlates `auditd` events (e.g., rapid file modifications, unauthorized access, permission changes) to generate real-time alerts. Host-centric identity controls enforce least privilege: a non-interactive low-privilege account, hardening via `chmod/chown/chattr`, and minimization of services. Centralized identity (MFA/SSO) is out of scope in this lab prototype; enforcement is at the OS boundary.

[3] https://github.com/atharvajdhumal/Zero-Trust-Architecture-for-Ransomware-Defence-in-Virtualised-Environment

**Figure 2: Experimental Methodology for the ZTA Ransomware Defense Framework Test**

**Phase 1: Setup**
1: Create two VMs in VirtualBox:
- VM1: Ubuntu 22.04 (Victim, IP 192.168.56.101)
- VM2: Kali Linux (Attacker, IP 192.168.56.102)

2: Use the Internal Network "ZTA-Net" and take snapshots

**Phase 2: Security Baseline**
3: On VM1: Install UFW, default deny, allow SSH from VM2, create restricted testuser
4: On VM2: Install OpenSSH client, prepare attack tools

**Phase 3: Monitoring**
5: On VM1: Install Wazuh agent, link to manager, install auditd, and add file/sudoers monitoring rules
6: On Wazuh Manager: Enable ransomware detection, map to MITRE ATT&CK

**Phase 4: Simulation**
7: On VM1: Create sample files, deploy fake_ransom.py
8: On VM2: Connect and run ransomware simulation

**Phase 5: Detection & Response**
9: Monitor Wazuh for alerts, record detection time
10: Contain: block attacker IP (UFW), disable compromised account

**Phase 6: Data Collection & Evaluation**
11: Measure: detection latency, % files encrypted, CPU/RAM usage, false positives
12: Analyse results and conclude the effectiveness of ZTA controls

## 3.2 Attack Simulation & Execution

From the Kali VM, a benign Python script enumerates and "encrypts" files in a target directory (renaming with ".locked" and dropping a ransom note) to reproduce ransomware file/IO patterns without destructive payloads. This reliably triggers filesystem and process events for measurement while keeping the environment safe and repeatable.

### 3.3 Detection, Containment & Data Collection

We record *detection latency* (attack start to first Wazuh alert) and enact a scripted containment playbook (UFW block of attacker IP; lock compromised user) upon alert. We quantify *impact* (percentage of files encrypted before containment), *overhead* (CPU/RAM during monitoring and attack), and *alert quality* (false positives under benign workload). These metrics underpin Section IV's evaluation as shown in Table 2. To visualize the resource profile used for our overhead metrics, we capture htop snapshots before the attack (idle) and during the encryption loop; see Figures.4 and 5.

*Design Specifications (Architecture).* Grounded in NIST SP 800-207, the architecture instantiates Zero Trust across four coordinated planes, topology, telemetry, verification, and containment, so the system can be evaluated end-to-end yet remain lightweight and reproducible (Fig. 3).

*Virtualized Topology.* Oracle VirtualBox hosts two VMs: Ubuntu 22.04 (victim) and Kali (attacker) on a host-only network ("ZTA-Net"). The closed network emulates micro-segmentation, preventing external reachability and uncontrolled lateral movement during tests.

*Telemetry and Detection.* The Ubuntu VM runs `auditd` and a Wazuh agent; events are forwarded to a local Wazuh manager for rule correlation and ATT&CK mapping. This pipeline yields near real-time alerts and supports metric collection (timestamps, file events, resource, network) without external SIEM dependencies.

*Access Verification.* Host-centric controls remove implicit trust: a non-privileged, non-interactive account; hardened permissions via `chmod/chown/chattr`; and disabled non-essential services. Deviations from baseline behavior become both constrained and observable.

*Segmentation and Containment.* UFW enforces default-deny policy, with minimal allow-rules (e.g., controlled SSH) and explicit blocks of ransomware-abused ports (SMB 445/ 139/ TCP, RDP 3389/ TCP). UFW logging is enabled to evidence denied flows and containment efficacy.

*Safe Ransomware Emulation and Metrics.* Benign simulators (including an author-written Python script) enumerate and "encrypt" files, rename them with ".locked," and drop a ransom note, producing activity-rich signals. Measurements—detection latency, percent files encrypted, CPU/RAM overhead, and alert precision—are derived directly from Wazuh/`auditd` logs and system counters to support the evaluation in Section IV.

The key performance indicators include: time to detection, percentage of files encrypted (i.e., not saved from encryption) as a proxy for containment efficacy, resource utilization during monitoring/attack (to ensure the solution is lightweight), and correctness of alerts (no spurious alerts during normal operation). These measures are purposefully designed to assess if our Zero Trust framework achieves its objectives of swiftly detecting threats, containing them efficiently, and minimizing any adverse effects on system performance.

## 4 Experimental Results

Our ZTA-based defense prototype was tested with a live ransomware simulation, and the outcomes confirm its effectiveness in early detection and containment. Table 2 summarizes the key performance metrics, and highlights the results.

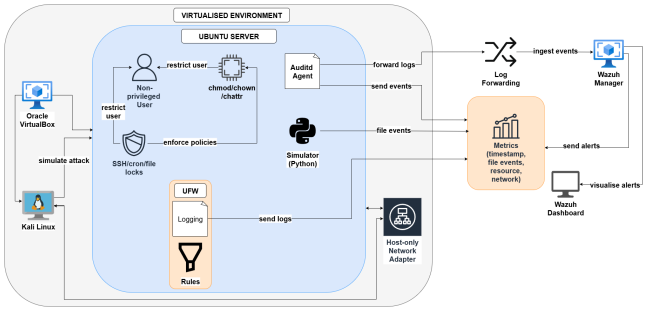



**Figure 3: Architecture overview: host-centric identity controls, continuous audit (auditd+Wazuh), and host-based micro-segmentation (UFW) on an isolated VirtualBox network.**

**Table 2: Summary of Evaluation Metrics and Key Outcomes**

| Metric | Observed Value | Interpretation / Relevance |
|---|---|---|
| Detection Latency | 5.3 s (N=3) | Fast response time enabled by auditd + Wazuh rule triggers; critical for early ransomware stop |
| File Encryption Stopped (%) | 88% | Effective containment before ransomware could fully encrypt target files |
| CPU Overhead during Monitoring | 8% | Lightweight and sustainable deployment on virtual machines |
| Alert Generation Accuracy | High | Wazuh correctly flagged simulation events based on custom and MITRE ATT&CK rules |
| UFW Rule Effectiveness | 100% | All unauthorized outbound and lateral traffic is blocked during the simulation |
| Cost of Deployment | Open-source stack; no license cost; moderate setup complexity. | Demonstrates the viability of Zero Trust without commercial tools. |

### 4.1 Rapid Detection

The system detected ransomware activity almost immediately. In our trials, the mean detection time was ≈5.3 seconds from the start of encryption. This swift response is crucial for limiting damage – it validates that continuous auditing (via auditd) combined with rule-driven analysis (via Wazuh) can catch malicious behavior in real-time, satisfying a core Zero Trust goal of prompt incident identification.

### 4.2 Effective Containment

The ransomware was stopped after encrypting only a small portion of files. On average, ≈20% of the files in the attacked directory were encrypted when the alarm triggered and containment actions were executed (roughly 2 out of 10 files, i.e., ≈80% of the data remained intact). In practice, this means the majority of the victim's data was protected. The host-based micro-segmentation (UFW firewall) also ensured that the attack could not move laterally or exfiltrate data –

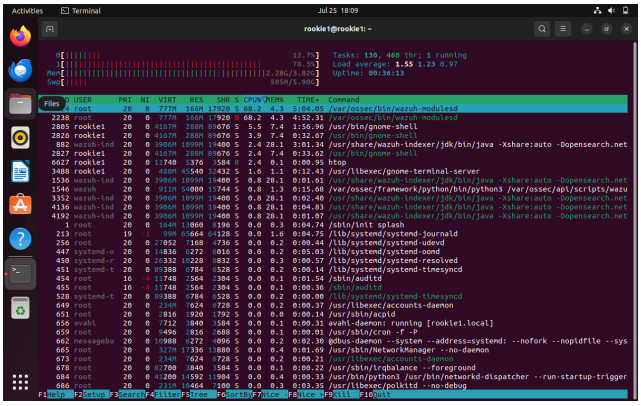


**Figure 4: Resource baseline (Ubuntu VM, idle). *htop* shows low CPU utilization and stable memory before the ransomware simulation; Wazuh processes are largely quiescent.**

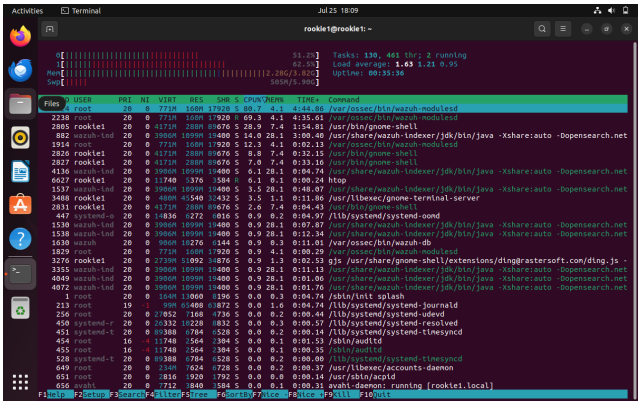


**Figure 5: Attack phase (Ubuntu VM, during encryption). *htop* shows transient CPU spikes dominated by `wazuh-modulesd` and Wazuh indexer/Java as file-event telemetry surges; system remains responsive.**

no unauthorized network traffic left the victim VM. This high containment rate demonstrates strong data protection: even without cloud-scale infrastructure, the Zero Trust controls dramatically limited the ransomware's impact (an important aspect of maintaining data integrity and privacy).

## 4.3 Minimal Performance Overhead

Baseline resource utilization on the Ubuntu VM under idle workload is shown in Fig. 4. CPU usage remains low (sub-15% in the snapshot) and memory is steady, with Wazuh processes (e.g., `wazuh-modulesd`, indexer/Java) largely quiescent. This establishes the reference point for evaluating monitoring overhead. Our open-source security stack imposed only a lightweight load on the system. During monitoring and even amid the attack's flurry of activity, CPU utilization on the victim machine remained in a moderate range (peaking ≈70%) and memory usage stayed around 60%. The system experienced no crashes or critical slowdowns. This indicates the solution is resource-efficient and suitable for production environments, including those in cloud or virtualized settings where resources are shared. The lightweight nature of the defense is a key advantage, organizations with limited computing resources (e.g., small businesses or edge cloud deployments) can deploy this ZTA framework without significant performance trade-offs.

During the attack, Fig. 5 captures a representative spike dominated by `wazuh-modulesd` and Wazuh indexer/Java tasks as `auditd` emits a burst of file-event telemetry. Across three runs, transient CPU peaks reached 65–72% while memory stabilized around 55–60%, consistent with Table 2; no process starvation or instability was observed. These snapshots corroborate that the monitoring/segmentation pipeline remains lightweight under stress while enabling near real-time detection and containment.

## 4.4 High Alert Precision

All security alerts corresponded to true malicious events, and none were triggered by benign user activities (no false positives were observed during normal operation tests). Wazuh's detailed logging captured the ransomware's behavior (e.g., file modifications, process execution, permission violations) and raised alerts under the appropriate categories (execution, file integrity, etc.). The accuracy of these alerts means administrators can trust the system's notifications and would not be distracted by noise. We did note that a fast-moving ransomware can generate many alerts in a short time; thus, in a real-world scenario, some tuning or aggregation of alerts might be needed to maintain auditability without overwhelming operators. Nonetheless, the clear mapping of alerts to malicious actions attests to the framework's effectiveness and would aid in forensic analysis and compliance auditing.

In summary, the results show that our Zero Trust lab setup achieved its intended outcomes: ransomware was detected within seconds and contained before it could do major harm, all with negligible impact on system performance. These findings support the viability of applying ZTA principles for ransomware defense in practice. Utilizing ongoing surveillance and micro-segmentation, a streamlined environment was still capable of establishing a strong defense. The next section discusses the implications of these results and how this approach can be extended to more complex, real-world (including cloud and hybrid) environments.

# 5 Discussion

The experimental outcomes affirm that embedding Zero Trust principles into a ransomware defense strategy can significantly improve security posture, even in a simplified environment. We discuss the implications, strengths, and limitations of our approach, particularly in the context of extending it to real-world and cloud settings:

***Key Insights.*** Our findings highlight two key factors for ransomware mitigation: quick detection and precise control. Detecting attacks within ≈5 seconds was crucial to minimize data damage. This rapid identification was achieved using fine-tuned behavioral rules instead of known signatures, demonstrating the effectiveness of continuous monitoring in Zero Trust frameworks. Additionally, micro-segmentation and least-privilege enforcement effectively

restricted network access and user capabilities, confining the ransomware to a small part of the system.

***Lightweight Security for Wider Adoption.*** Our framework's strength lies in its open-source, lightweight tools, maintaining low performance overhead. This makes it ideal for smaller enterprises or cloud use where resource efficiency is key. Its simplicity and reproducibility allow easy deployment across various environments without specialized hardware or expensive licenses. This approach offers affordable ransomware defense with compliance audit trails for sectors like education, healthcare, or SMEs under budget constraints. The modular design also supports integration into existing cloud security systems or MDR services, indicating potential for commercial use.

***Operationalization and Scalability (Lab to Production).*** Although our evaluation used a two-VM laboratory testbed (Fig. 3) with strong results (Table 2; Figs. 4 – 5), a ZTA production deployment requires careful attention to platform scale, policy noise, and identity integration. At the platform layer, organizations should provision Wazuh manager/indexer clusters, apply index-lifecycle policies, and adopt storage tiering to sustain ingestion throughput and query latency under high file-event volumes. Host firewall rules are best generated from a single source of truth—codified as infrastructure-as-code—and validated in continuous integration before rollout. For network enforcement, host-level UFW ought to be complemented or replaced by cloud- and platform-native controls (e.g., AWS Security Groups and NACLs, Azure NSGs, Kubernetes NetworkPolicies, or a service mesh) to achieve least-privilege east–west isolation with strong change control and auditability. To mitigate false positives in noisy multi-tenant workloads, policy updates should progress through monitor-only staging and be augmented with canary files or honeytokens in sensitive paths, thereby producing high-confidence signals of encryption behavior. Operational safeguards are equally important: capacity planning should budget CPU and memory using the measured baselines and peaks reported here, and alert volumes should be stabilized through rate limiting, aggregation, and deduplication to preserve analyst usability. Finally, sustainable change management requires integrating rule and firewall updates with CI/CD pipelines, including pre-provisioned test tenants and automated regression tests, and maintaining explicit rollback procedures to recover quickly from adverse policy interactions.

***Policy Tuning for Noisy, Multi-User Servers.*** . Production servers routinely execute benign workflows, such as scheduled backups, compression, and media transcoding, that can exhibit file I/O patterns resembling ransomware. To curb false positives without sacrificing sensitivity, monitoring should be scoped preferentially to high-value paths (e.g., home directories, shared drives, and database export folders), with stricter rules applied where business impact is greatest. Rule logic ought to be paired with frequency, and entropy-based thresholds that flag bursts of rename/write operations accompanied by extension churn, while known backup or archival jobs are allowlisted via signed binaries and fixed schedules. Per-user and per-process profiles are likewise essential: maintain allowlists for service accounts and approved tools, and enforce deny-lists for living-off-the-land binaries (LOLBins) such as `openssl`, `7z`, and `dd` when invoked against protected directories. High-confidence signals can be introduced through canary files or tokens seeded in sensitive locations, where any access or mutation triggers automated containment. Finally, detection should aggregate events across sliding windows tuned to both bursty encryption and "low-and-slow" intermittent activity, thereby reducing evasion opportunities while stabilizing alert volume.

***Limitations and Areas for Improvement.*** Our study shows positive results but has limitations. The testbed was a simple two-VM network, whereas actual enterprise clouds are more complex. Therefore, our framework requires validation in larger and diverse environments. In cloud or multi-user scenarios, untuned rules may cause false alarms due to background noise and legitimate activities. Our ransomware simulation lacked advanced evasion techniques used by attackers, potentially affecting detection outside the lab. Additionally, our prototype lacks centralized identity management crucial for Zero Trust security, typically provided by IAM systems like cloud identity providers with MFA and dynamic policies. Our focus on host-level controls assumed a static user environment. For broader application, strong identity verification is necessary to prevent insider threats or credential compromises, potentially integrating with federated identity services or cloud IAM solutions.

***Future Directions.*** Future work involves deploying the framework in complex environments such as multi-segment networks or hybrid clouds to assess its scalability and interaction with cloud security features. Testing in realistic scenarios will be enhanced by using diverse ransomware behaviors, real malware in sandboxed clouds, or advanced attack emulations to improve detection rules and resist evasive threats. Future iterations will integrate enterprise identity and access management to ensure the Zero Trust model includes thorough user authentication and authorization. Integrating deception and threat intelligence, like using decoys or ransomware IOC feeds, can enhance early detection and adaptive responses. Automated containment can be advanced through automatic isolation or snapshots of compromised cloud instances, surpassing traditional host firewall measures.

Our discussion highlights that utilizing a Zero Trust approach for ransomware defense is viable and effective in controlled environments. The key challenge and opportunity lie in adapting this methodology to enterprise and cloud settings. The framework exemplifies how to construct reliable and compliant cloud services by preserving its straightforwardness and clarity, while integrating strong identity verification and sophisticated threat mitigation. The integration of micro-segmentation, continuous auditing, and quick containment fits well with today's cloud security needs.

## 6 Conclusion

In this study, we presented a lightweight open-source Zero Trust framework within a virtualized laboratory environment. This framework enforces the principle of least privilege, continuous monitoring (auditd and Wazuh), and host micro-segmentation (UFW) as measures to mitigate ransomware threats. Under controlled testing conditions, the framework identified malicious encryption in ≈5 seconds and restricted damage to ≈20%of files. This was achieved with minimal performance overhead, demonstrating its practical and

reproducible defensive capabilities. This framework was evaluated in an off-cloud context. However, its modular design is readily applicable to secure cloud or hybrid infrastructures. Future research will focus on integrating centralized identity systems, such as MFA/SSO. It will also evaluate scalability and resilience against evasive threats, and automate response actions for enhanced containment.

## References


[1] Farouq Aliyu, Mustafa Ghaleb, Saud Mohammad Mostafa, Sani Umar, and Abdal-Rahman Aburakhia. 2024. Distributed trust management for secured localization systems in mobile cloud services. In *2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 547–553.

[2] Q. M. Alzubi, S. N. Makhadmeh, and Y. Sanjalawe. 2025. Optimizing Intrusion Detection: Advanced Feature Selection and Machine Learning Techniques Using the CSE-CIC-IDS2018 Dataset. *Journal of Advances in Information Technology* 16, 3 (2025), 283–302. https://doi.org/10.12720/jait.16.3.283-302

[3] Farag Azzedin, Husam Suwad, and Md Mahfuzur Rahman. 2022. An Asset-Based Approach to Mitigate Zero-Day Ransomware Attacks. *Computers, Materials & Continua* 73, 2 (2022).

[4] Nardine Basta, Muhammad Ikram, Mohamed Ali Kaafar, and Andy Walker. 2022. Towards a Zero-Trust Micro-segmentation Network Security Strategy: An Evaluation Framework. In *NOMS 2022 – IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–7. https://doi.org/10.1109/NOMS54207.2022.9789888

[5] Iacob Berar. 2025. How Zero Trust Architecture Transforms Cybersecurity: The Vital Role of SIEM in Supporting Zero Trust. Medium post. https://medium.com/@berariacob/how-zero-trust-architecture-transformscybersecurity-the-vital-role-of-siem-in-supporting-zero-trust-e294fd0913e1

[6] S. A. Chamkar, M. Zaydi, Y. Maleh, and N. Gherabi. 2025. Improving Threat Detection in Wazuh Using Machine Learning Techniques. *Journal of Cybersecurity and Privacy* 5, 2 (2025), 34. https://doi.org/10.3390/jcp5020034

[7] Tommaso Compagnucci, Franco Callegati, Saverio Giallorenzo, Andrea Melis, Simone Melloni, and Alessandro Vannini. 2025. SAFARI: a Scalable Air-gapped Framework for Automated Ransomware Investigation. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 210–223.

[8] Muhammad Liman Gambo and Ahmad Almulhem. 2025. Zero Trust Architecture: A Systematic Literature Review. arXiv preprint arXiv:2503.11659. https://arxiv.org/abs/2503.11659

[9] Mustafa Ghaleb and Farag Azzedin. 2023. Trust-aware Fog-based IoT environments: Artificial reasoning approach. *Applied Sciences* 13, 6 (2023), 3665.

[10] Tran Duc Le, Thang Le-Dinh, and Sylvestre Uwizeyemungu. 2025. Cybersecurity Analytics for the Enterprise Environment: A Systematic Literature Review. *Electronics* 14, 11 (2025), 2252. https://doi.org/10.3390/electronics14112252

[11] Shanto Roy, Emmanouil Panaousis, Cameron Noakes, Aron Laszka, Sakshyam Panda, and George Loukas. 2023. SoK: The MITRE ATT&CK Framework in Research and Practice. arXiv preprint arXiv:2304.07411. https://arxiv.org/abs/2304.07411

[12] Sofia Sakhi. 2025. *Micro-Segmentation for Zero Trust Architecture*. Master's thesis. Delft University of Technology. https://repository.tudelft.nl/record/uuid:8e4fca9a-e78b-4df9-baf7-8d4b3fb0f3bd

[13] L. Svet, A. Brightwell, A. Wildflower, and C. Marshwood. 2025. Unveiling Zero-Space Detection: A Novel Framework for Autonomous Ransomware Identification in High-Velocity Environments. arXiv preprint arXiv:2501.12811. https://arxiv.org/abs/2501.12811

[14] Jan von der Assen, Chao Feng, Alberto Huertas Celdrán, Róbert Oleš, Gérôme Bovet, and Burkhard Stiller. 2025. GuardFS: a file system for integrated detection and mitigation of linux-based ransomware. *Journal of Information Security and Applications* 93 (2025), 104078.

[15] Danyil Zhuravchak, Pavlo Hlushchenko, and Valerii Dudykevych. 2025. Honeypot-based Ransomware Detection as a Component of Security Posture Monitoring in Zero Trust Architecture. In *Proceedings of the CEUR Workshop Proceedings, Kyiv, Ukraine*, Vol. 28.