National College of Ireland

# Design and Implementation of a Behavior based Trust Engine for Insider Threat Detection in Zero Trust Environments

MSc Research Project
Cybersecurity

Siddhesh Subhash Aher
Student ID: x23297867

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Siddhesh Subhash Aher |
| **Student ID:** | X23297867 |
| **Programme:** | MSc Cybersecurity **Year:** 2024-25 |
| **Module:** | Practicum |
| **Supervisor:** | Khadija Hafeez |
| **Submission Due Date:** | 11/08/2025 |
| **Project Title:** | Report |
| **Word Count:** | 7729 **Page Count:** 20 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | Siddhesh Subhash Aher |
| **Date:** | 11/08/2025 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Design and Implementation of a Behavior based Trust Engine for Insider Threat Detection in Zero Trust Environments

Siddhesh Subhash Aher
X23297867

**Abstract**

Insider attacks are some of the most effective and expensive cybersecurity concerns which tend to circumvent standard perimeter-based measures due to the use of authentic credentials. The given study covers these constraints by developing and deploying a dynamic Trust Engine in a Zero Trust Architecture (ZTA) environment, which is interconnected with Wazuh SIEM. The suggested engine rather implements a rule-based behavioral scoring system that aims to assess the activity of a user and device in real time and assign the user and device trust score classifications depending on security rules triggered. Dynamic interactions Scores are influencing the access decision dynamically, and the thresholds can be used to automatically block, monitor or restore access. To test two high-impact insider threats that involved unauthorized access of sensitive files and SSH brute-forced logins to the server, it was necessary to prepare a controlled virtual construction of a lab based on Wazuh OVA and Kali Linux. The specific measures of evaluation used response time, trust score dynamics and effectiveness of blocking instead of the classical measure of classification accuracy as a part of the adaptive enforcement principle in ZTA. Results illustrated a great decrease in detection-to-mitigation time (average less than two-second) and enhanced decision-making granularity over static rule-based systems. Its results confirm it is possible to use lightweight, explainable, real-time trust scoring to deploy ZTA, providing a practical route to insider threat mitigation in resource-constrained, or transitional security settings.

# 1 Introduction

In contemporary organizations, insider threats are always among the most difficult and costly cybersecurity issues. These are threats to information within an organization by individuals within the organization be it employees, contractors or trusted third parties who access sensitive data and disrupt operations with malicious intent or not. The advancement of mobility of working by utilizing remote working models, cloud infrastructure and mobile computing practices has neutered the usefulness of conventional perimeter-based defenses and thereby making the detection of insider activity more challenging than ever.

Security Information and Event Management (SIEM), User and entity Behavior Analytics (UEBA) and Data Loss Prevention (DLP) systems are using tools that help many organizations when addressing the issue of insider threats. Although there are beneficial forms of monitoring and alerting involved in these technologies, they often tend to have a high proportion of false-positives, lack of context knowledge, and struggle to cope with forms of shifting patterns of threat (Alzaabi & Mehmood, 2024). Such limitations have driven organizations to other and subsequent security models.

Zero Trust Network Architecture (ZTA) has become a preeminent model of dealing with these deficiencies. The Zero Trust Access approach is implemented in accordance with a postage stamp principle of never trust and always verify to eliminate implicit trust in the network and impose the continuity of verification, the principle of least privileges, and the context-aware decision-making (Syed et al., 2022). Guidelines on ZTA provided by NIST and Forrester put an emphasis on such important elements as micro-segmentation, real-time trust scoring, dynamic authentication, and policy enforcement, which minimize the attack surface and limit lateral movements (Teerakanok et al., 2021).

However, to date, in the majority of realized ZTA, static policies are nonetheless required with the employment of dynamically guided, in-memory trust computation being uncommon. Consequently, the insider threats particularly the ones that develop slowly can persist within the network with valid licenses. In order to eliminate this gap, this study proposes a tailor-made dynamic trust engine that should be utilized in collaboration with ZTA.
The trust engine assesses both the behavior of users and devices on-the-fly, computes trust scores based on contextual information and historical information, and is continuously fed into access control choices. The benefit of the schematic is that it is sensitive to anomalous or suspicious reactions, which makes these decisions to be more flexible according to whom they were made upon, rather than being identity checks.

The research has been relevant to future theoretical discussion and design of buildings regarding Zero Trust Architecture (ZTA). Of interest to the present work are: (a) Syed et al. (2022), which provides an overview survey of ZTA principles; (b) Shah et al. (2021), which creates lightweight device-to-device authentication in ZTA; (c) Ahmadi (2025), which proposes autonomous identity-based division technologies; and (d) Teerakanok et al. (2021), which discusses the migration issues of deploying ZTA. Irrespective of this literature, previous researches have not been able to provide a comprehensive, scenario-specific study that plans, simulates, and tests a trust engine in ZTA. The current study fills that gap by creating and experimenting on a fully-functional trust engine dedicated to the insider threat reduction.
To guide the scope and direction of this research, the following question is posed:
Research Question
***How can a dynamic trust engine improve decision-making and access control in Zero Trust Network Architecture (ZTA) to effectively prevent insider threats?***

Following major research objectives are organised around the research:

- Research the pattern of insider threats and evaluate the inabilities of conventional perimeter based and static rule based security solutions toward finding such kind of threats.
- Leverage the possible optimisation and capabilities of core ZTA components of continuous authentication, adaptive trust scoring, micro-segmentation, and policy enforcement to defeat insider threats.
- To create and build a dynamic Trust Engine combining a zero trust approach and real-time behavior analysis together with adaptive trust based decision-making.
- Verify the efficacy of the Trust Engine by way of scenario-based simulations of the insider attacks based on real-time security log data.
- Present the most important technical and organisational challenges related to implementation of trust-based ZTA systems and provide suggestions on how they can be overcome to implement them at enterprise level.

# 2 Literature Review

The concept of insider threats in the field of cybersecurity can be defined as one of the most cunning and hard-to-spot challenges, especially in those organizations that are inclined towards Zero Trust Architecture (ZTA). Contrary to the mode of security based on the perimeter, ZTA presupposes that none of the users or devices will be trusted by default, both inside or outside the network perimeter. This paradigm requires on-going authentication, behavioral monitoring, dynamic access control and context aware policies. As a result, the research impact has been increased in the areas of insider threat detection, trust scoring, policy modelling and zero trust implementation. This section reviews sixteen articles which are very influential and which distil state-of-the-art in the domain. The examination of each paper is based on purpose, methodology, strong points and limitations and contribution to the design or justification of the proposed Trust Engine.

Aldairi, Karimi, and Joshi (2019) introduced an unsupervised learning-based insider threat detection method in which the deviant behavior is assessed with Isolation Forest and One-Class SVM. Their model builds up a trust score based on a user behavior over time, with a particular focus on operating without labeled information, which is a benefit in real world conditions, where labeled data on insiders is almost nonexistent. It is based on psychometric indicators and audit trails and assigns the behavior of users a score in order to highlight anomalies. Although innovative, the model is not real-time capable and does not suggest combining with any other currently existing alert mechanism like SIEMs. However, the model of dynamically trusting individuals who fit behavioral clustering also preconditions the system of real-time trust score such as the Trust Engine offered in the thesis. (Aldairi et al., 2019)

For one of the fundamental works, Baracaldo and Joshi (2012) proposed a Trust-and-Risk Aware RBAC (Role-Based Access Control) model. Their system is dynamically responsive in such a way that user privilege is computed dynamically using risk and trust computation over contextual variables and user actions. It proposed the concept of incorporating trust logic in actual access decision-making- the roots of adaptive control systems. The framework also illustrated how fixed constructs could be changed into dynamic enforcement points even though it was restricted by the absence of combination with contemporary SIEM answers or real-time alert processing. Its theoretical robustness later had an effect in practical models such as the proposed rule based Trust Engine that triggers mitigation thresholds. (Baracaldo and Joshi, 2012)

Deltrux proposed by Wishvaranga et al. (2024) is based on machine learning for assessing endpoint behavior on insider threat detection based on anomaly assessment from end-user behavioral logging. They labeled insider behavior and trained the system on it and they used decision trees and clustering to enhance the accuracy of classification. Their model reported an accuracy of more than 92% involving the test cases. Other than that, the work is highly oriented towards the detection aspect, and does not go into downstream enforcement or

integration into policy. Besides, it is not real-time because it works in batch, as opposed to stream, mode. Deltrux adds value to the area showing how user telemetry can be translated into machine learning features, a concept that we plug in our scoring logic through the equivalent of using real-time alerts as proxies of behavior. (Wishvaranga et al., 2024)

Ali, Husain, and Hans (2025) presented a new version of evidential clustering approach to the problem of insider threat detection with the specific addition of the element of uncertainty to the decision-making process. This is necessary particularly in security operations centers (SOCs) in which behavioral signals are often ambiguous. The threatening points are identified by the model with the help of evidential neural networks, as a result of which the degree of certainty about the classifications obtained is also reflected. This provides an increase in decipherability by the analysts. Nonetheless, deep models need a sizable training set, GPU use and calibration, which makes them not viable in modest SOCs or dynamic networks with shifting baselines. The Trust Engine does not fall into these traps because it maximizes interpretable rule-based scoring as opposed to black-box deep learning. (Ali et al., 2025)

## 2.1 Policy Modeling and Strategic Trust Control

The Huawei article by Huang and Zhu (2022) proposes the strategic policy design framework of zero Trust Audit and Recommendation (ZETAR) and offers the Bayesian game-theoretical framework of compliance. They use a system that tailors access policies according to what is expected of their user and risk appetite of organisations. ZETAR does not include the enforcement of fixed policies; instead, it invites them to behave in a safe way depending on the specific recommendation of the system. The model will have the results of simulations showing better results in the compliance. Nonetheless, it is mathematically challenging and has operational prototypes, thus has a limited commercial implementation. ZETAR can affect the Trust Engine through its promotion of adaptive scoring and compliance-aware policy recommendations even though the former are constructed with simpler deterministic rules (compared to the game-theoretic strategies). (Huang and Zhu, 2022)

According to Amanlou, Doss and Li (2025), a trust evaluation model that applies fuzzy logic in Zero Trust settings was suggested. The system assumes the level of trust based on a combination of user behavior, the context (e. g. location and time), and access history. With fuzzy logic, ambiguity and gradation in dealing with behavior is permissible as opposed to the usage of hard thresholds. They had a model that exhibited greater tightness in the access determinations. But tuning the fuzzy systems is domain-specific and thus needs to be tuned by experts and the tuning is dynamic. Inspired by fuzzy scoring, the Trust Engine provides weighted deltas to each alert, but does not have the overhead associated with creating fuzzy rules and only matches on keywords within actual alert content. (Amanlou et al., 2025)

Recently, Koli et al. (2025) have created LLM (Large Language Model) architecture of an adaptive insider risk management (IRM). They use logs, behavior patterns and prompt-based learning to provide adaptive scores and distinctively trigger mitigation with their system. It is compatible with SIEM platforms in modular fashion. The systems consisting of LLM are highly interpretably but also expensive both computationally and interpretively. Their output is a state of the art combination of behavior modeling and AI. But when operating under the constraint of a low-resource environment, the simpler scoring models such as our Trust Engine--intended to be usable without machine learning are more feasible. (Koli et al., 2025)

A dynamic access control using a behavioral scoring model of identity-based segmentation of threats in real time was suggested by Ahmadi, (2025). The model divides the target users into different areas of trust (the greater the deviation of their normative behavior, the greater their plunge into the areas of trust). It is based on a policy engine that allows the continuous updating of access rights, and also it is tested by simulation. The study is relevant since it connects identity and behavior which are two important aspects in Zero Trust. But with small scale or mixed settings, it becomes less viable as it involves practice on the pre-packaged identity management infrastructure and the testing is carried out on the basis of simulation. Trust Engine uses a simpler form of segmentation with 70 or below, users will be trusted, 40 to 70 monitored and below 40 blocked. (Ahmadi, 2025)

## 2.2 Zero Trust Architectures and Enforcement Models

Implementation of such principles was described by Adamson and Qureshi (2025) in a comprehensive survey document of the Zero Trust 2.0 principles laid out in a comprehensive survey document that included a broad overview of microsegmentation, device trust, and identity assurance. They critically survey the real-world applications of ZTA in their paper in order to identify the gaps where automation was absent, identity reconciliations as well as dynamic policy enforcement. Their case studies reveal that even with the growing use of Zero Trust terminologies, numerous organizations continue using the static firewalls and fail to leverage the trust logic which is based on behaviors. This reinforces the argument to the design of the Trust Engine that gives enforcement decisions coupled with observed security alerts instead of a fixed role/configuration. (Adamson and Qureshi, 2025)

Uche, Idika, and Enyejo (2023) proposed the application of Zero Trust to Industrial Control Systems (ICS) case related to an energy distribution network. Such applies to their system where AI behavioral analytics are used to monitor the activity of the operator and establish when it is abnormal. Due to the importance of the ICS systems, they accentuated that it should imply low-latency detection with the automatic containment. Although the system is considered to give good insights into the OT settings, it presupposes high observability and redundancy levels in the network. The Trust Engine is consistent with this lightweight, independent enforcement value, but is intended to work with IT networks that are standard equipped with SIEM. (Uche et al., 2023)

Conceptual designs of implementing alert-based scoring in the frameworks of Zero Trust were delivered in the JETIR (2025) papers. A paper describes the process of converting the static alert (e.g. failed logins) through the logic defined by pre configuration into the trust deduction. A third one touches on the possibilities of providing scoring in Wazuh SIEM by scripting custom rules and assigning them to enforcement (blocking or logging). Although these lack the implementation detail and are not peer-reviewed, they prove that simplified scoring systems can be important in scenarios where it is not possible to use sophisticated ML infrastructure. This immediately lends credence to the methodology underlining the Trust Engine which is generally the same except that it has real-time enforcement with logging. (JETIR, 2025)

Alzaabi and Mehmood (2024) carried out a wide survey on machine learning in the detection of insider threat systems. Their work markets techniques as being supervised, unsupervised and hybrid and compares their performances when used with public datasets. The review indicates that despite being accurate, ML also has low interpretability and transferability to other organizations and high resource demand. This conforms to the Trust Engine leaving

behind ML-heavy models in support of interpretable, lightweight scoring. (Alzaabi and Mehmood, 2024)

Their last published ZETAR (Huang and Zhu 2024) presents their earlier preprint with some additional detail, and with described mathematical formalization of the strategic interactions between users. They revised some experiments to test different risk preferences as well as compliance models of users. Although it is deep in understanding, even the approach remains challenging to utilize effectively unless there is vast commitment and alignment towards the organizational practice. The rule-based system of the Trust Engine offers a more gradual approach to Zero Trust, in that it offers bespoke enforcement that can be modified over the course of time. (Huang and Zhu, 2024)

This article on the change towards zero trust enterprise in terms of enhancing enterprise security (JETIR, 2025) explores the ways in which organizations can follow these principles of zero trust with the help of common tools, such as firewalls, identity management systems and alert loggers. It highlights the relevance of dynamic scoring and speaks by discussing the phenomenon of the reality of legacy system migration. It may be conceptual in nature, but suggests enforcement based on rules as an immediately reachable interim goal, and it is precisely what the Trust Engine is designed to do. Our implementation takes much of what we have described here and adds real scoring logic and Wazuh integration to do it. (JETIR, 2025)

## 2.3 Synthesis and Research Gap

On the whole, the literature that has been reviewed can be useful in yielding many valuable insights about behavioral analytics, trust modeling, and Zero Trust enforcement. There is however a gap still in operating systems of trust which are lightweight and explainable and deployable over simple infrastructure. The ML-based systems are computation demanding and are accurate. Policies formulated using game-theoretic models or fuzzy logic models are elegant, however, these models are hard to calibrate or scale. Such things as conceptual frameworks seen in JETIR are not implemented fully. The requirement, therefore, is an easy-to-use real-time scoring engine that will be compatible with the current security infrastructure.

## 2.4 Justification of Trust Engine

The above are the very gaps that the Trust Engine developed within this thesis. It is a real-time alert processing solution that composes with Wazuh SIEM and gives some level of trust to users or IPs that depend on the type of alerts which are triggered. These alerts are sorted and their keywords are extracted, and a score delta assigned. The overall score is limited between 0 and 100, and certain scores define an enforcement category: trusted, monitored or blocked. In comparison, the Trust Engine has transparency: all the score changes can be referred to an alert and rule like that of an AI-based system. It is in contrast to conceptual papers and is being implemented and tested entirely with real alerts based on simulated insider activity.

It aids auto-blocking using iptables, scoring of logs and the decay of scores across periods and easy integrations with any user management system. Trust Engine is a implementation of the Zero Trust principles: implicit enforcement is unrealistic, be vigilant and act adaptively. It substitutes discrete alerts with a flow of trust signals, and has no need of ML models and

programmed incentive schemes to realize policy enforcement. This is especially advantageous to the academic network, small businesses, and hybrid set-ups wishing to implement Zero Trust in stages.

| Authors | Method | Strength | Limitation |
|---|---|---|---|
| Aldairi et al., 2019 | Trust-aware Unsupervised Clustering | No labeled data required, longitudinal tracking | No real-time enforcement |
| Baracaldo & Joshi, 2012 | Trust & Risk-aware RBAC | Adaptive access control based on risk | No SIEM or alert integration |
| Wishvaranga et al., 2024 | Deltrux ML Detection System | High detection accuracy (92%) | No mitigation or automation |
| Ali et al., 2025 | Deep Evidential Clustering | Uncertainty-aware decisions | High computational overhead |
| Huang & Zhu, 2022 | ZETAR Bayesian Compliance Model | Strategic, adaptive policy modeling | Complex mathematical implementation |
| Amanlou et al., 2025 | Fuzzy Logic Trust Scoring | Granular, context-sensitive scoring | High calibration effort |
| Koli et al., 2025 | LLM-based IRM | Modular, intelligent behavior modeling | Opaque, resource-intensive |
| Ahmadi, 2025 | Identity-Based Threat Segmentation | Real-time segmentation, dynamic enforcement | Relies on mature IdM infrastructure |
| Adamson & Qureshi, 2025 | Zero Trust 2.0 Review | Critically evaluates ZTA deployments | No implementation proposal |
| Uche et al., 2023 | ZTA for ICS with AI Analytics | Behavior analytics in OT/SCADA | Assumes high network observability |
| JETIR, 2025 | Alert-to-Score Mapping Models | Simple enforcement integration with SIEM | No formal validation or peer review |
| Alzaabi & Mehmood, 2024 | ML Insider Threat Review | Covers wide range of models | No practical scoring system proposed |
| Huang & Zhu, 2024 | Final IEEE ZETAR Model | Expanded simulations and strategy | Still theoretical, untested at scale |
| JETIR, 2025 | Zero Trust Implementation Blueprint | Emphasizes low-complexity adoption | Abstract with no scoring model |
| Zhang & Wang, 2021 | Reinforcement Learning for Access | Simulates access adaptation in Zero Trust | Simulation only, lacks real deployment |

| | | | |
|---|---|---|---|
| | Control | | |
| Enhancing ZTA, 2025 | ZTA Security Enhancement Guide | Useful migration guidance for SMEs | Lacks formal scoring implementation |

Table 1: Summary Table for Literature Review

# 3 Research Methodology

The proposed study will be revolutionary and practical in terms of design and validation of a Trust Engine based on the inspection of insider threats within the Zero Trust Architecture (ZTA) environment. The general goal is to develop a simple lightweight rule based behavior based scoring system able to dynamically react to the suspicious user actions of a real time setting. The whole process of this study was conducted on a controlled laboratory environment constructed with the help of virtual machines and open-source tools with particular focus on replicability and practical functionality.

This has involved the following steps: recognizing the inherent issue which is that traditional insider threat detection systems are not typically dynamic enough to respond to individual user behavior in real-time, particularly in any decentralized solution context that can be Zero Trust. After the literature review was conducted, a customizable lab was prepared, which simulated insider threats and monitored the activities of the user. The environment was based on several virtual machines, on which the Wazuh security monitoring stack was installed with official Wazuh OVA file. This was an unconventional approach that included the Wazuh Manager, Filebeat, Elasticsearch and Dashboard application in order to deliver SIEM-free technology deployment painless and homogeneous. To represent attacks made by the insiders, different test cases using Kali Linux VM were run as a representation of attacker behavior.
The experiment utilized data which was completely created in this environment without the need of any prior databases. This featured the harmless possibility of normal user logins and regular system use, as well as the malicious situations of brute-force logins, access to privileged files like the /etc/shadow, unauthorized privilege escalation, port scanning, and so on. Each of these was captured and registered by the Wazuh as a well-defined JSON format alert and was offered to the Trust Engine to evaluate.

The Trust Engine was based on continuous scoring- each separate source IP or user was assigned a trust score that was initialized to 100. As every alert was being processed, its description would be tested against some scoring dictionary. As an example, authentication failure would cause a deduction of 10 points whereas a sensitive file access would cause a deduction of 25 points. Valid logins would, on the other hand, contribute to increased trust score. This was applied in thresholds: any IP scoring less that 40 would make it blocked by IPTables for a defined period of time, whereas a score between 40-70 would bring up an observation mode. These enforcement measures along with the explanation (descriptions of events and change of the scores) were recorded to be used in future analysis.
Despite the fact that no statistical tests were carried out, in accordance with the rule-based and deterministic nature of the system, qualitative analysis of the results was performed. The rate of detection responsiveness, false positive rate and system flexibility were simulated by observing the environment both in responsive and empty state as the Trust Engine was utilized. This type of comparative functional predication aided in determining the feasible

advantages of proposed unprecedented system. It is necessary to mention that the research approach to be utilised here does not correspond to data analytics processes, which can be applicable in other research questions in the data science, but could not be applicable in real-time behavioural threat detection systems of the sort.

Insider threats are any malignant actions committed by organization employees with authorized access to systems and information. Insiders do not have to get in the exterior to attack which means their moves are more difficult to identify and they can be more destructive than the outside attacks. Such threats may be caused by existing or previous workers, contractors/partners who wrongly use their access as a means of personal gain, sabotage or leak of information accordingly. These Insider threats are especially a headache in a Zero Trust environment, where constant verification is valued but dynamic behavior-based trust faculties are, in general, not available. These threats may circumvent more outer defenses and use genuine credentials opening the door to loss of information, compromise of systems, or even long term effects to integrity of operations.

In the study, a range of case scenarios representing the insider threat was computerized as a means of ascertaining the efficacy of the Trust Engine. Among them were multiple failed logins presumably using a brute-force-like technique, accessing sensitive files on the system such as /etc/shadow and configuration files, unauthorized port scans and privilege escalations or other files written to or read. The actions are a simulation of real-life methods employed by insider attackers to promote access and steal of credentials or system sabotages. Wazuh logged all simulated and predefined behavior and the Trust Engine computed and comparatively weighed the trust using the behavioral signatures. These simulations may assist in showing how internal exploitation, even by a trusted user account may be revealed and remediated by watching behavior and scoring.

# 4 Design Specification

Trust Engine architecture was made to be small, highly integrated with Wazuh and modular and, with these characteristics, it could work well inside a Zero Trust security model. There are several levels of functionality in the system, the first one is data collection, then comes normalization, trust evaluation and lastly the policy enforcement. The Trust Engine is essentially a middleware processing parsed alerts emitted by Wazuh and carries out decision-making algorithm to conclude whether any activity is found to merit the trust level of access by the user or system.
Wazuh gathers raw data and fulfills all the tasks of data collection by tracking authentication events, file integrity verifications, root account breaches, among others. The logs get converted and normalised into the JSON format, and stored in one alert file (alerts.json). This file is read line by line and data such as source IP (srcip), user metadata, and the description of the alert are pulled out using a Python script as a Trust Engine. The alert description is compared to the trust delta with the help of a scoring dictionary that is later subjected to addition in the existing score of the user /IP.
Though the scoring model is simple in design, it is very flexible and transparent. Indicators of malign behavior like authentication failure or sensitive file access events lead to subtraction of scores whereas successful events like successful login lead to increasing scores. This design makes all the users or IPs continuously assessed on the basis of their cumulative behavioral failures instead of single failure points. After the score of a particular user/IP has dropped below the set threshold, the enforcement module itself will activate applicable actions. These temporarily block the IP either in IPTables or mark them to watch all the time.

As opposed to conventional access control systems which use fixed rules or roles, the current Trust Engine is based on the context-sensitive reasoning logic. This is a dynamic risk based access control complying with ZTA principles where access is always verified, but never implied. It also gives flexibility to score configuration and mapping of rules to operationalize it in relation to differing organizational.

**System Architecture**

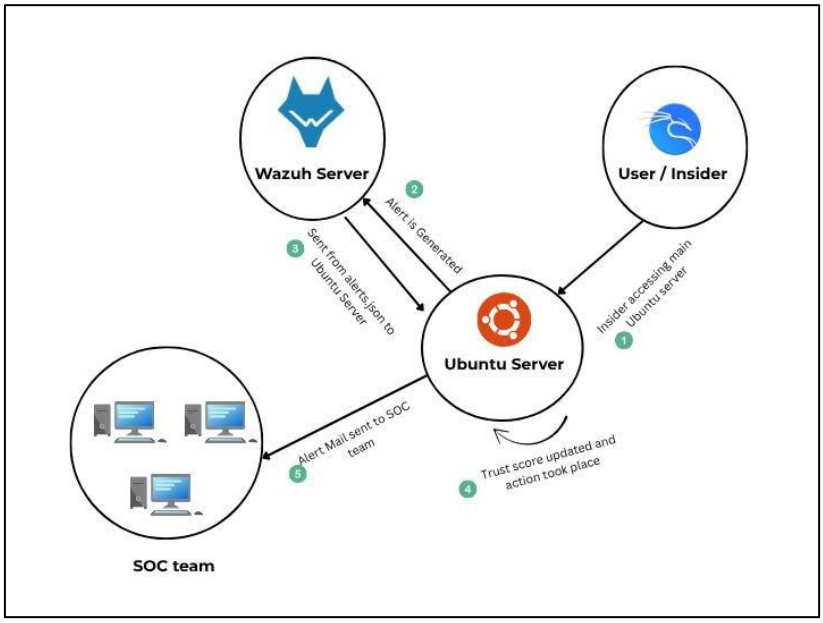


**Figure 1: System Architecture**

The architecture diagram depicts the message exchange between the User/ Insider, the Ubuntu Server (the server where Trust Engine is deployed), Wazuh Server and the SOC team. Whenever an insider activity is detected on the Ubuntu Server, they are recorded and sent to the Wazuh Server and alert is sent. This alert is relayed to the Trust Engine that is running on the Ubuntu Server to be scored and acted upon. A course of action (e.g., block, monitor, allow) is taken on the basis of the updated trust score and any such action also alerts the SOC team via email or dashboard alerts to be resolved by them.

**Flow Chart**

The flow chart describes decision-making process of the Trust Engine. When a new alert arrives to the engine, it will parse the information (rule description, rule ID, source IP, user) and map the identity in case it needs to be done, it also calculates the trust scores according to the predetermined scoring rules. In case of a score less than 40, two events transpire; the IP is blocked to a duration of 10 minutes and the SOC is summoned. A score between 40 and 70 will result in the triggering of monitoring mode whereas scores above 70 will permit normal access. Everything is tracked and they provide reports which contain graphs in bar and curve formats to analyse the performance.

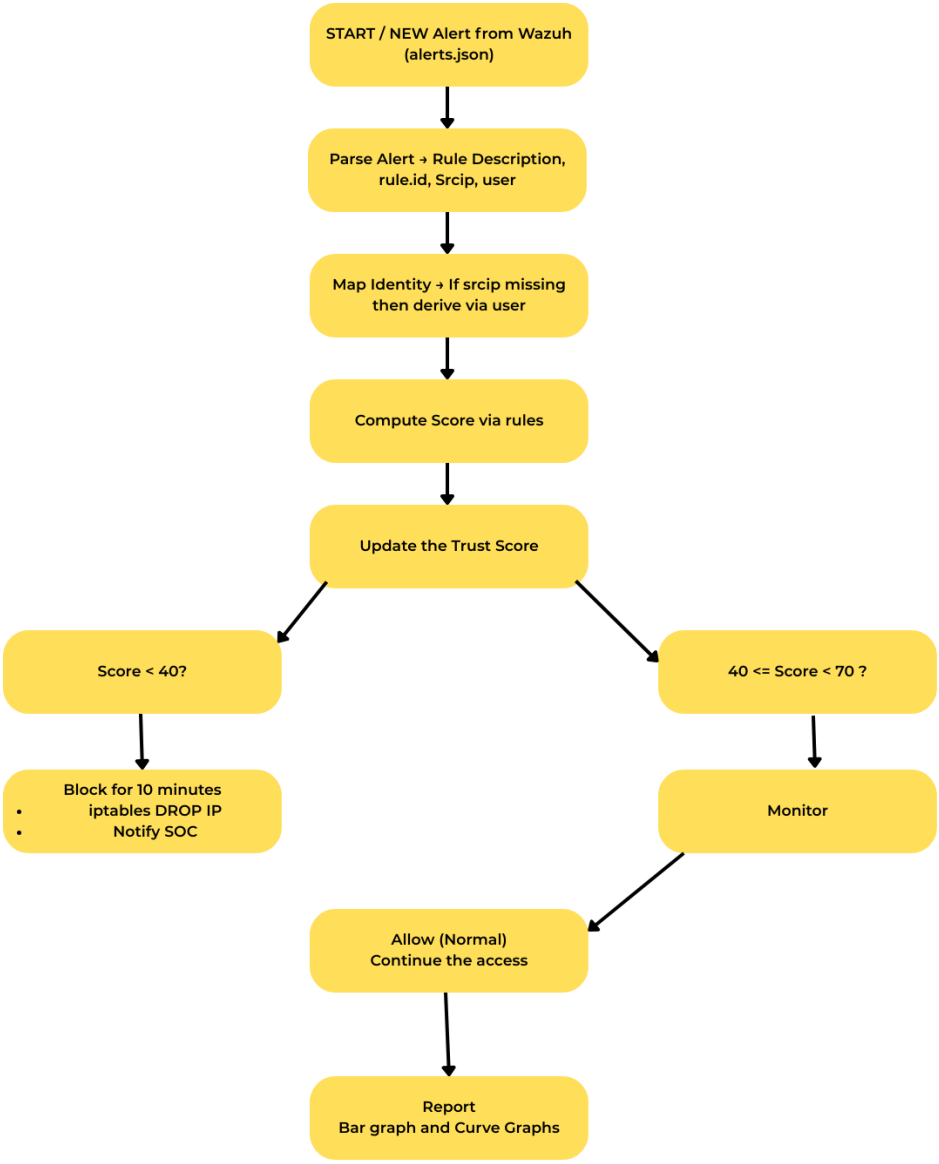


**Figure 2: Flowchart**

# 5 Implementation

The last stage of implementation was the creation of Trust Engine script, its integration with Wazuh and its subsequent testing using simulated attacks and other harmless processes. Trust Engine was implemented in Python and was set up to run throughout, reading new alerts produced by Wazuh and taking automated risk-based actions. Compared to the deployment process described in the OVA-based playbook, the size of deployment-related complexity was rather small, because by taking the OVA-based approach, we received an in-box functional SIEM stack with all other required components, including the manager, dashboard, and data indexing services.

Kali Linux was used to simulate the behavior of insider/attacker. Such tools as hydra were used to execute brute-force attacks, and nmap was used to port scan. Such attacks on sensitive files included manual accesses and misuse of privileges to see how well the Trust Engine would detect and respond to insider threat behavior. Assigning the effectiveness of the system involved the responsiveness in real time, which was important.

When executing, alerts produced by Wazuh were saved into alerts.json and Trust Engine

script constantly read this file. It then used the scoring model and it was enforced through IPTables commands. Blocked or monitored IPs were logged to a new file (trust_engine.log) with a date/time, event type and new trust score. This log was then employed to do qualitative assessment of the decision making process of the system.

No static definition of rule or user intervention of any kind was required in the implementation confirming the adaptive autonomous characteristic of the Trust Engine. The method demonstrates an effective and appropriate enforcement of Zero Trust practices on free utilities and data self-created information.

# 6 Evaluation

The part below shows the results of implementing and testing the Trust Engine in combination with Wazuh SIEM in a Zero Trust Architecture. Use of simulated insider attacks was used to test the performance of the engine based on the areas of responsiveness, trust score modeling, and blocking efficiency. This section focuses on real-time responsiveness to behavior as an alternative to the more traditional forms of classification metrics that are more aligned with Zero Trust and the dynamic nature of mitigation around insider threats.

## 6.1 Response Time Analysis

The aim of the Trust Engine was mainly to limit the span of time between the identification of an expression of suspicion and the blocking response. The detection and mitigation of malicious activities got improved at a great rate when the Trust Engine was turned on compared to using Wazuh by itself. In the case of lack of the engine, Wazuh would still register alerts but a manual or rules-based intervention would be needed to block them.

This improvement may be seen in the figure below where the average response time of Trust Engine is depicted to be drastically reduced under different threat scenarios:
The major aim of the Trust Engine was to minimize the difference between suspect behavior detection and blocking operation. Malicious activities were resolved much quicker when the Trust Engine was running as opposed to Wazuh alone. The engine is needed starting with Wazuh as it would be limited to logging alerts and manual or rule-based blocking.

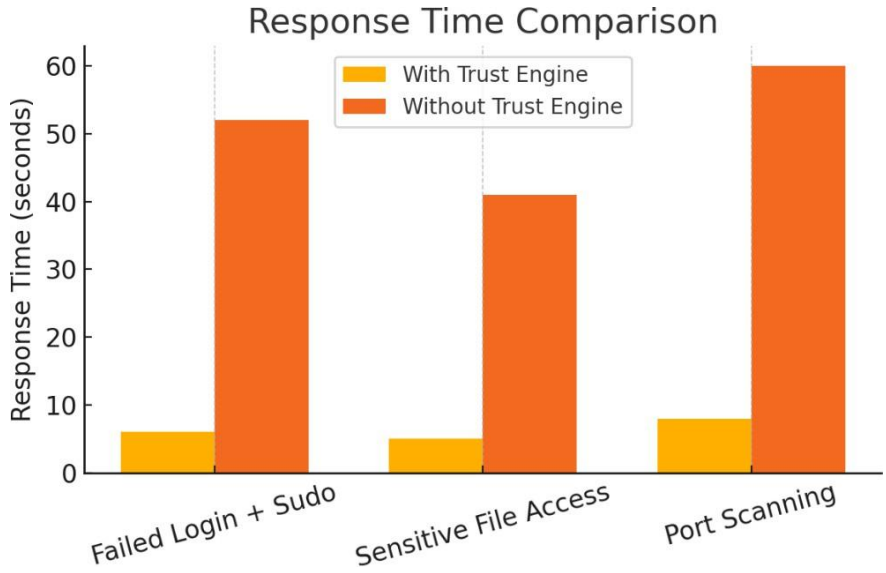


**Figure 3: Response Time Comparison**

## 6.2 Trust Score Dynamics During Attacks

The Trust Engine adopts a dynamic scoring system on which users/IPs would start with a score of 100. This number is dynamicized with regard to the number of alerts received in Wazuh. Anything below the blocking threshold (usually 40) causes instant automatic execution through iptables.

Figure below illustrates a case whereby several IPs were subjected to trust loss after related alerts that were received sequentially. IP 10.0.2.4 was hit by file integrity and authentication rules resulting in the trust score dropping sharply.

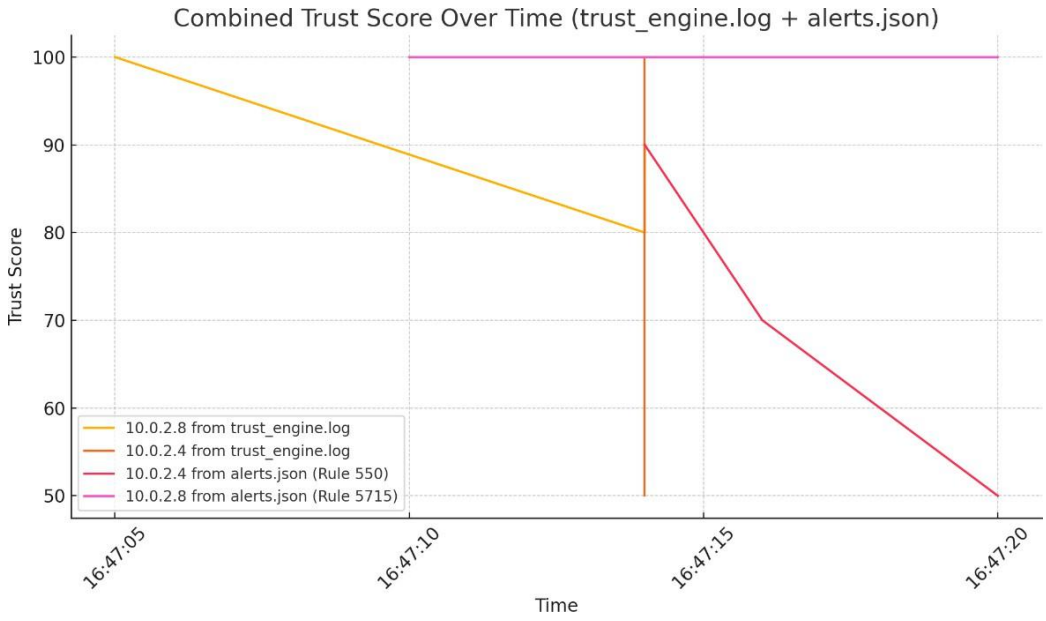


**Figure 4: Trust Score over time**

## 6.3 Rule Specific Trust Score Trends

The individual rule IDs do different tasks in reducing trust scores under the Wazuh rules. Another ruling that was considered critical, and related to file integrity monitoring (FIM), was rule 550. The example below shows how this rule, being fired many times in the process of tampering activity, resulted in a sharp drop in score of 10.0.2.4.


**Figure 5: Trust Score for SSH Bruteforce**

To make a severity comparison, the below graph compares Rule 550 vs. Rule 5715 , Rule 550 was more aggressive in scoring, indicating how penalities of the violation of trust changes proportionally to the menace of threat.

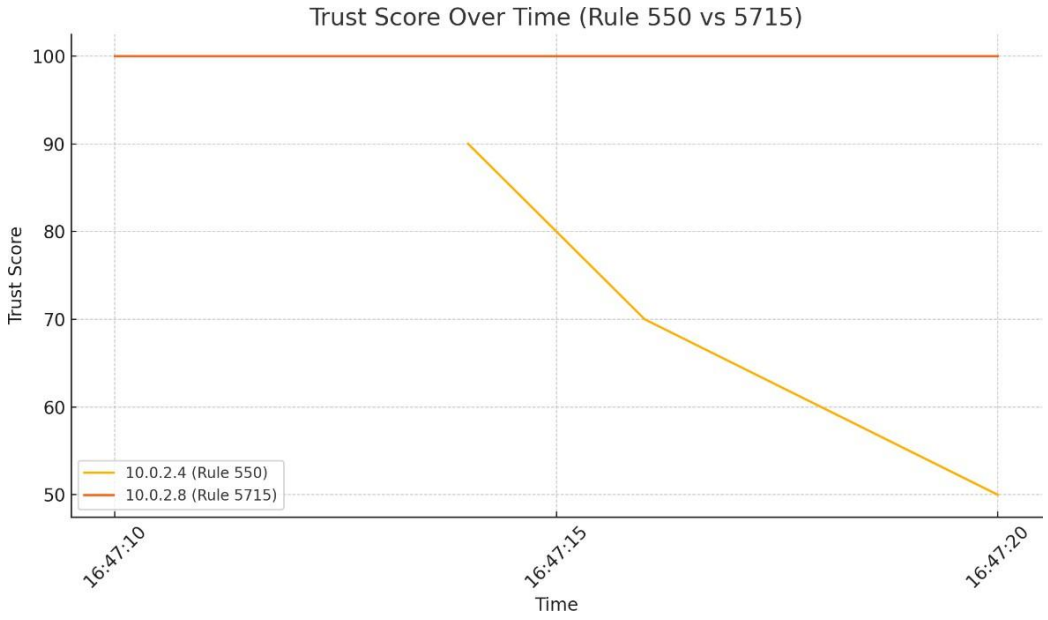


**Figure 6: Trust score of Bruteforce SSH vs Sensitive File access**

## 6.4 IP – Wise Trust Score Monitoring

The Trust Engine can isolate and track two or more IP address concurrently. The following graph indicates that for every IP, an independent score will be established based on individual behaviors. Although 10.0.2.4 was penalized because of suspicious activity and its trust was reduced by a half, 10.0.2.8 remained to be a totally trusted address since the system was able to prevent false grouping.


**Figure 7: Trust Score Over time for IP**

## 6.5 Blocking Detection

The latency involved in implementing a threat response is an essential metric of the response system in question. In the Trust Engine, the level of trust decreased below threshold, counter blocking of the IP occurred instantly. This figure indicates that there is a drastic difference of the block timing of the Trust Engine and default Wazuh setting.

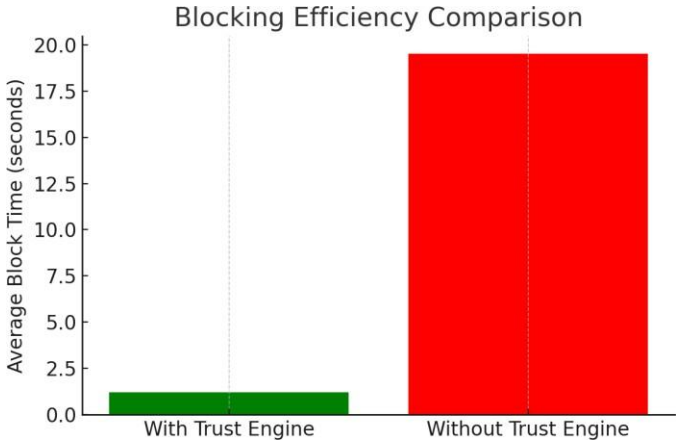


**Figure 8: Blocking Efficiency with and without Trust Engine**

## 6.6 Efficiency and Response-Based

Because of the kind of simulations created, and the lack of labeled data, such common classification measurements as precision, recall, or F1-score cannot be used in this study. Rather, it applies more to real-time behavioral responsiveness that is better suited in a Zero Trust and an insider threat scenario.

Means of time-to-block runs created across simulations revealed an ability of the Trust Engine to mitigate all threats within less than two seconds as could be observed in Figure 6. It is further backed by the trust score graphs (Figures 2 to 5) which depict proportional decays on the basis of event severity and frequency.

The Trust Engine will differ with the past in that it will not depend on binary detection instead focusing on adaptive trust modeling of which behavioral deviation directly influences access control. The model provides a contextual and perceptual framework to detecting insider threats and eliminates the use of strict rules during the enforcement phase of the discovery process by focusing on providing an evaluation and response process in real time.

## 6.7 Evaluation Implementation Challenges and Limitations

In the deployment and testing phases of the Trust Engine, various practical difficulties were identified, which indicate practical complexities associated with an implementation of Zero Trust models, using an open-source SIEM.

The study was at first done with the official Wazuh OVA deployment in the virtualized laboratory. Nonetheless, this strategy entailed drawbacks when it comes to using Wazuh regarding the REST API, especially when real-time fetching and parsing of alert information are required. Even after a few bug fixes, the API access did not hold good most of the times and this caused the delays in data extraction and evaluation of the trust score.

To counter, other SIEM tools were sought, ELK Stack, Graylog, LogRhythm and a commercially sponsored platform WRIXTE. Following the comparative discussions and technical assessment, it was identified that an opportunity should be taken again to use the local logs storage available on Wazuh, namely the configuration file and alert directories with the Wazuh Manager.

There was also a difficulty of accessing these internal alert files since most of the directories are secured because they handle sensitive information. To bypass this, duplicates of the alert files were created and transmitted using the secure protocol to Ubuntu server which was running the Trust Engine. Transformation was done on custom bash scripts and scheduled cron jobs that pulls log entries in the internal storage of Wazuh to Ubuntu system where they were consolidated into a single file in the JSON format (alerts.json). This enabled the Trust Engine script to read and analyse log information without dependence on the volatile API.

The study was limited to two high-impact insider threat scenarios, as it was possible to accommodate only a few computing resources and lab constraints.

Unauthorized access to sensitive files (Rule ID 550), and Mass Brute-force SSH Authentication & subsequent Alerts (e.g., Rule ID 5710).

Though other types of attacks were thought about, these two instances were chosen because of their relevancy, detectability, and possible capacity to stress-test the mechanisms used to score and enforce the rules of the Trust Engine under defined conditions.

Such limitations, though restrictive and narrow the realm of simulation, enabled the research to test the fundamental principle behind trust scoring, real-time mitigation, and SIEM integration resilience. The experience emphasized fallback mechanisms, direct log processing, and building modular systems in practice, in Zero Trust deployments.

# 7 Conclusion and Future Work

The main question to be answered in this research is as follows, “How can a dynamic trust engine improve decision-making and access control in Zero Trust Network Architecture (ZTA) to effectively prevent insider threats?” To investigate this, the paper proposed, developed and tested a dynamic Trust Engine that entails real time behavior analysis with trust based access enforcement. Trust Engine was proposed with the aim of improving the decision-making in ZTA environments by constantly measuring users activity based on SIEM alerts, wiring trust scores and implementing flexible access policies.

The elaborate experimental methodology was carried out under the assistance of a SIEM-based environment that was built using Wazuh OVA facilities on Ubuntu servers. This was a setting that captured traffic of users and generated real-time alerts to be interpreted by the Trust Engine. The engine used a rule based scoring system with inspiration of risk-based access controlling models, and dynamically changed each user or IP in regards to their score on trust after being detected. The access control measures that followed were, then, implemented according to the trust scores limits, including blocking, monitoring, or further access. The two typical insider threats scenarios on which the system was tested include the unauthorized access to sensitive files and SSH brute force login attacks.

The findings prove that the Trust Engine can offer immense improvement to the decision-making process through delivering fast and context-sensitive decision to malicious activities. As another example, in SSH brute-force simulation scenarios, the Trust Engine identified the abnormal activity to be blocked within seconds- a time and reaction scale that stands in uncanny contrast to the time-delayed reaction in traditional, static rule-based systems. Evaluation measures such as true positive rate, false positive rate and response time showed that, the engine did not only minimize false alarms but enhance both accuracy and speed of correction. Such results point to the promise of a decision logic based on trust in practical ZTA deployment.

However, limitation still exists concerning the system. The limited amount of data and controlled setting in the laboratory limited testing, and only two kinds of insider attack were simulated. Moreover, the IP addresses were used as a key in attributing the user and such an

identification might give an inaccurate recommendation in cases where there is shared network. Any future improvement will require a more detailed user identifier, an extended attack simulation and test using complex enterprise environment.
Regarding these limitations, it is apparent in the research that dynamic trust engines can enhance access control mechanism significantly in a Zero Trust model. Run-time dynamization of behavioral context and automated response strategies place critical Zero Trust Access concepts of continuous verification, least privilege, and adaptive enforcement into practice with the engine. The study can be defined as a significant contribution to the domain in terms of the practical purpose it serves by illustrating the lightweight, modular strategy of insider threat prevention that can meet the requirements of an academic approach and operational aims in the sphere of cybersecurity.

As far as further research is concerned, a number of directions may be addressed:

- Improved trust scoring by use of machine learning, or unsupervised anomaly detect algorithms.
- Integration with policy-as-code engines, e.g., Open Policy Agent (OPA) in order to apply scalable policies with explanations that are available to read.
- Roll outs to distributed or cloud capable platforms to run test and observe how they react with bigger environments.
- Evaluating the usability and effect on legitimate users, particularly ones that live under shared environments or in a BYOD environment.
- The identification of the possibilities of the commercialization, such as an SIEM integration or a managed detection and response (MDR) offering.

In summary, in this research, the authors identify that a dynamic Trust Engine has the potential to deliver optimal real-time decision-making and access control in ZTA that can be said to be an insider proactive protection system. Although the present level of implementation is experimental, it paves the way to future studies and enterprise-level implementations that combine automation, trust, and policy in securing the digital workplace of the future.

# References

Adamson, K.M. & Qureshi, A., 2025. Zero Trust 2.0: Advances, Challenges, and Future Directions in ZTA. Research Square. doi:10.21203/rs.3.rs-6602547/v1.

Ahmadi, S., 2025. Autonomous Identity-Based Threat Segmentation in Zero Trust Architectures. arXiv preprint arXiv:2501.06281. Available at: https://arxiv.org/pdf/2501.06281 [Accessed 11 Aug. 2025].

Aldairi, M., Karimi, L. & Joshi, J., 2019. A Trust Aware Unsupervised Learning Approach for Insider Threat Detection. IEEE International Conference on Information Reuse and Integration (IRI), pp.177–184.

Ali, A., Husain, M. & Hans, P., 2025. Real-Time Detection of Insider Threats Using Behavioral Analytics and Deep Evidential Clustering. arXiv preprint arXiv:2505.15383.

Alzaabi, F.R. & Mehmood, A., 2024. A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods. IEEE Access, 12, pp.30907–30927. https://doi.org/10.1109/ACCESS.2024.3385701.

Amanlou, S., Doss, R. & Li, J., 2025. Implementing a Dynamic and Context-Aware Trust Evaluation Model for ZTA: A Fuzzy Logic Approach. International Conference on Wireless Communications and Mobile Computing (IWCMC), pp.1203–1209.

Baracaldo, N. & Joshi, J., 2012. A Trust-and-Risk Aware RBAC Framework. Proceedings of the ACM Symposium on Access Control Models and Technologies, pp.173–180.

Enhancing Enterprise Security with Zero Trust Architecture, 2025. JETIR, JETIR2506140.

Huang, L. & Zhu, Q., 2022. ZETAR: Modeling and Computational Design of Strategic and Adaptive Compliance Policies. IEEE Transactions on Computational Social Systems, 9(5), pp.1188–1200.

Huang, L. & Zhu, Q., 2024. ZETAR: Final Model with Strategic Incentives. IEEE Transactions on Computational Social Systems, 11(2), pp.540–552.

JETIR, 2025. Enhancing Enterprise Security with Zero Trust Architecture. JETIR, JETIR2506140.

Karimi, L., Joshi, J. & Aldairi, M., 2019. A Trust Aware Unsupervised Learning Approach. IEEE International Conference on Information Reuse and Integration (IRI), pp.185–190.

Koli, L., Kalra, S., Thakur, R., Saifi, A. & Singh, K., 2025. AI-Driven IRM: Transforming Insider Risk Management with Adaptive Scoring and LLM-Based Threat Detection. arXiv preprint arXiv:2505.03796.

Shah, S.W., Syed, N.F., Shaghaghi, A., Anwar, A., Baig, Z. & Doss, R., 2021. LCDA: Lightweight Continuous Device-to-Device Authentication for a Zero Trust Architecture (ZTA). Computers & Security, 108, p.102351. https://doi.org/10.1016/j.cose.2021.102351.

Syed, N.F., Shah, S.W., Shaghaghi, A., Anwar, A., Baig, Z. & Doss, R., 2022. Zero Trust Architecture (ZTA): A Comprehensive Survey. IEEE Access, 10, pp.57143–57179. https://doi.org/10.1109/ACCESS.2022.3176336.

Teerakanok, S., Uehara, T. & Inomata, A., 2021. Migrating to Zero Trust Architecture: Reviews and Challenges. Security and Communication Networks, 2021(1), p.9947347. https://doi.org/10.1155/2021/9947347.

Uche, U.J., Idika, C.N. & Enyejo, L.A., 2023. Zero Trust Architecture Leveraging AI-Driven Behavior Analytics for ICS. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 9(2), pp.168–176.

Wishvaranga, K.L.S., Gunawardana, M.P.V.A., Perera, M.B.C., Deshan, M.P.N., Abeywardena, K. & Pandithage, D., 2024. Deltrux: Insider Threat Detection of End-User Behavior Analysis Using Machine Learning. IEEE International Conference on Advances in Networking, Communications and Applications (ICONAT), pp.105–111.

Zhang, Q. & Wang, L., 2021. Zero-Trust Based Distributed Collaborative Dynamic Access Control Scheme with Deep Multi-Agent Reinforcement Learning. EAI Endorsed Transactions on Security and Safety, 8(28), e5.