Using Zero Trust Principles for Detecting Authorization Attacks in Cloud Environments

Ivan Parkhomenko[1,†], Larysa Myrutenko[1,†], Roman Ohiievych[1,*,†] and Mykhailo Savonik[1,†]

[1] *Taras Shevchenko National University of Kyiv, Bohdana Havrylyshyna St. 24, Kyiv, Ukraine*

Abstract

An underlying tenet within the zero-trust architecture is the statement of never trust, always verify, and therefore is strongly applicable for securing cloud environments that often lack defenses at the perimeter. The distributed nature of cloud infrastructures, the ability to dynamically scale resources, and complex access patterns render them particularly vulnerable to authorization attacks. In this paper we discuss how employing Zero Trust principles – including but not limited to continuous identity verification, least access to any resource, and micro-segmentation – can allow for better detection capabilities plant to authorization attacks in cloud environments. In this paper, we propose a framework utilizing real-time monitoring and other machine learning algorithms to detect abnormal behavior from this server which would suggest an attack of this nature is taking place. Our study shows that using Zero Trust strategy improves authentication threats detection and mitigation significantly via those simulations and empirical tests. These findings provide important information that can be used to strengthen cloud security frameworks and mitigate vulnerability to authorization attacks.

Keywords

Zero Trust Architecture, Authorization attacks, Cloud security, Anomaly detection, Identity verification, Machine learning algorithms, Access control

1. Introduction

Cloud services are revolutionizing the information technology arena as they offer scalable, stateless and on-demand access to a shared pool of configurable computing resources [1]. Cloud services allow organizations to improve operational effectiveness, cut costs, and speed up innovation [2]. As per Gartner, worldwide spending on public cloud services would grow to $332.3 billion by 2021 as the dependency on cloud solutions rises. While these implementations offer plenty of advantages, they also present unique security challenges for cloud computing. A key issue here is the occurrence of attacks on authorization, such as unauthorized access, and privilege escalation attacks that can take advantage of weaknesses in access control to obtain unauthorized access to resources [3]. These risks are augmented by the nature of cloud environments which are distributed and dynamic [4]. An example of this is the 2019 Capital One data breach which compromised the personal information of more than 100 million customers and was attributed to a misconfigured web application firewall in the cloud, demonstrating the implication of authorization vulnerabilities [5].

Traditional perimeter-based security frameworks are no longer enough to combat these threats. These models assume that a trusted internal network is separated from an untrusted external network, an assumption that falls apart in cloud environments wherein resources are accessed over the internet and from different locations [6]. Thus, to properly address authorization attacks in the cloud configuration, more reliable security circuits have to be established to ensure accuracy regarding the nature of the user.

Information Technology and Implementation (IT&I-2024), November 20-21, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

ivan.parkhomenko@knu.ua (I. Parkhomenko); myrutenko.lara@knu.ua (L. Myrutenko); ohiievychr@fit.knu.ua (R. Ohiievych); savonikm@fit.knu.ua (M. Savonik)

0000-0001-6889-9284 (I.Parkhomenko); 0000-0003-1686-261X (L. Myrutenko); 0009-0003-7948-1125 (R. Ohiievych); 0009-0001-7622-978X (M. Savonik)

1.1. Zero Trust architecture

This brings us to ZTA or Zero Trust Architecture, a paradigm shift needed in in cybersecurity to overcome the limitations of the traditional security models. Based on the principle of outside-in security, Zero Trust ends the belief of trust in the network perimeter; the catch phrase "never trust – always verify" is rooted in this approach. This put the emphasis on verifying users and controlling access no matter where a user or network is located.

And when you think about cloud security, Zero Trust becomes truly relevant. The dynamic and distributed nature of cloud environments is conducive to Zero Trust principles, which are fundamentally about enforcing least privilege access and continuous authentication [9]. NIST has defined a robust Zero Trust architecture in their Special Publication 800-207, which can be incorporated into cloud infrastructures to improve security posture [7].

To address authorization attacks, cloud adoption of the Zero Trust model must be implemented so that access to resources is determined by rigorous verification and real-time risk evaluations. This solution solves the problems of multi-tenancy and elastic provisioning of resources by applying adaptive and contextual policies.

1.2. Objectives and contributions

The objective of this paper is to implement Zero Trust principles in order to create a strong authorization attack detection solution in cloud environments. The specific objectives are:

- To develop a mathematically rigorous model that allows one to combine Zero Trust principles for ongoing identity verification and access control in cloud-based environments.
- Machine learning algorithms: modeling anomalous behaviors as a sign of authorization attacks.
- To validate the proposed model with simulations and empirical analysis, so as to prove its relevance for improving detection mechanisms.

This study makes the following contributions:

- In this paper, we present a new model that combines Zero Trust principles with sophisticated mathematical models to enhance the detection of authorization attacks in cloud platforms.
- Real-time anomaly detection by machine learning based detection algorithms
- We evaluate the proposed solution and show how it outperforms traditional security models in detection accuracy and reduction of false positives.

This research is unique in its approach of using Zero Trust to talk on the most prominent issue of authorization attacks in cloud environments, hence adding one more piece in the puzzle of Zero Trust for academia and also providing some critical insights for the industry too. These findings can help strengthen cloud security frameworks and mitigate vulnerabilities arising from attacks targeting authorization mechanisms.

2. Background and related work

2.1. Authorization Attacks in Cloud Environments

Authorization attacks are a type of security breach in which an attacker defeats intended access control policies by obtaining access or escalating privileges within a system [9]. Such attacks in cloud environments can result in major data breaches, service disruptions and financial losses. Authorization Attacks Common examples include privilege escalation, session hijacking, and misuse of access controls due to misconfigurations.

Moreover, many cloud-specific attributes, such as the multi-tenancy of cloud architecture, dynamic resource allocation, and complex access patterns make the authorization attacks harder to detect in the cloud environment [10]. Because multi-tenancy allows several users/organizations to share the same physical resources, multi-tenancy increases the attack surface and possibilities for cross-tenant assaults [11]. One of the main features of cloud services is dynamic scaling, which means frequent alterations to the infrastructure that brings challenges to keeping security policies consistent.

In addition, there are complex access control requirements due to the use of distributed systems and microservices in the cloud [13]. Conventional security measures might not efficiently track or regulate the complex interplays among services, users and resources [14]. These complexities give attackers the opportunities they need to find holes in security configurations or bypass authentication and authorization.

In such environments, legitimate requests for access can be high-volume, and user behavior can vary widely, making it difficult to detect authorization attacks. The attacker has control over the normal user activity; therefore, normal-intended activity is similar to malicious user activity, which is hard for rule-based systems to identify legitimate activity from malicious activity [15]. In addition, due to the nature of the cloud as a service, latency requirements and performance are very important, which introduces limitations on the use of computing-intensive security measures.

2.2. Zero Trust principles

Zero Trust Architecture (ZTA) is a concept that works with a zero-trust approach - "never trust, always verify" - which removes any form of implicit trust. Key principles include:

- Continuous Verification: all access requests are continuously verified based on real-time context including user identity and device health.
- Least Privilege Access: users receive only the minimum level of access necessary, limiting potential harm to a compromised account.
- Micro-Segmentation: resources are partitioned into fine-grained areas to protect them from a lateral movement of an attacker.

For instance, Zero Trust was applied in the field of cybersecurity to provide better defense against sophisticated threats. One such example is Google's BeyondCorp [16], which relocates access control from the network perimeter to individual devices and users, with strict access control and continuous user authentication.

For cloud environments, Zero Trust concepts help manage the resource and user distributed nature. The adoption of continuous verification and least privilege access reduces the risk of such attacks on authorization systems [17].

Here are some methods for detecting authorization layer attacks:

- Rule-Based Systems: these systems employ predefined rules to identify unauthorized activities [18]. Known threats can be detected, unknown ones cannot [13].
- Anomaly Detection Techniques: they monitor user behavior to capture deviations from established baselines between users and normal behaviors [19]. Anomaly Detection Techniques: Machine learning algorithms which identify anomalies that might be evidence of future attacks.

But there are limitations to these approaches. To overcome that rule-based systems are updated frequently and lead to high false positives [18]. Anomaly detection solutions tend to fall short in dynamic cloud settings, and struggle to differentiate between benign deviations and malevolent behaviors [12].

They don't natively support Zero Trust principles, and as a result, existing mechanisms are often inadequate in cloud environments. Advanced detection methodologies must be paired with Zero Trust concepts to protect against these threats.

3. Problem statement

3.1. Traditional security models – the limitations

Traditional security models, largely focused on perimeter defense mechanisms, are failing to meet the cloud environment security challenges [7][8]. These models seem to hinge on the notion of a well-defined network boundary that separates trusted internal networks from untrusted external networks. On the other hand, this clear distinction is blurred in cloud computing as it involves the distributed nature of resources, virtualization and remote access capabilities [11].

Firewalls, intrusion detection systems, and network segmentation are the core elements of perimeter-based security [13]. While proven to be effective in on-premises infrastructures, such approaches do not suffice in cloud for the following reasons:

- Dynamic Resource Provisioning: the dynamic allocation of the resources of cloud services leads to constant changes in the network topology [12]. These changes are not necessarily static, with traditional security measures delivering only static protection that fails to adapt in real-time.
- Multi-tenancy: the sharing of physical resources among multiple tenants amplifies the attack surface and potential for cross-tenant attacks [10].
- Remote Access: cloud users access services from multiple locations and devices, where enforcing network perimeter security is impractical [14].
- Complex access patterns: The cloud being associated with microservices and in turn APIs are not accounted for in any traditional models, and such access patterns are very complex.

Table 1
Comparison of Traditional Security Models and Cloud Security Requirements

Aspect	Traditional Security Models	Cloud Security Requirements
Network Boundary	Clear internal vs. external	Blurred due to distributed resources
Resource Provisioning	Static	Dynamic and scalable
Access Patterns	Predictable	Complex and varied
User Location	Fixed, within organization	Remote and varied
Multi-Tenancy	Not applicable	Inherent characteristic

Table 1 presents limitations that underscore the inadequacy of conventional security solutions in overcoming cloud-specific challenges [11]. This exposes organizations to higher risks of authorization attacks that can go unnoticed beyond perimeter defenses [12][13].

3.2. Security with Zero Trust-based approach

The limitations of these traditional models reveal an urgent need for a security framework that is able to adjust to the dynamic and distributed nature of cloud computing. A Zero Trust Architecture (ZTA) provides such a framework by completely re-imagining the management of access and trust [7][15].

The limitations identified by are mitigated with Zero Trust principles:

- No Implicit Trust: since every access request is verified regardless of its origin, Zero Trust eliminates dependence on network perimeter.

- Real-time verification with Continuous Authentication and Authorization: Security policies withstand the impact of network conditions or user contexts.
- Fine-Grained Access Control: making use of least privilege access minimizes the effect of compromised credentials [15].
- Micro-Segmentation: splitting the network into smaller, controllable sections limits lateral movement by an adversary [16].

Zero Trust integrates additionally into cloud security, improving detection and protection against authorization attacks by:

- Better Insight: ongoing monitoring helps in having a better understanding of user actions and access patterns.
- Adaptive Policies: security policies can be adapted dynamically depending on contextual information [17].
- Improved Anomaly Detection: using Zero Trust and advanced analytics together enables detecting unauthorized access attempts [18].
- A Zero Trust-based Solution – proceeding on this, if organizations implement Zero Trust-based solution, they will develop a far more resilient security posture better fit for the realities of cloud environments.

4. Mathematical model and theoretical framework

4.1. System model

We formalize these components of the cloud environment relevant to our model in the following section. Users, resources, access requests, and security policies make up the system and are critical to establishing which assets the user can reach as well as enforcing Zero Trust.

Let:

- $U = \{u_1, u_2, \dots, u_n\}$ be the set of users.
- $R = \{r_1, r_2, \dots, r_m\}$ be the resource pool (e.g., data, applications, services).
- $A = \{a_1, a_2, \dots, a_k\}$ be the collection of permissible actions (e.g., read, write, execute).
- $S = \{s_1, s_2, \dots, s_p\}$ be the set of sessions.

An access request is defined as a tuple:

$$q = (u, r, a, s) \tag{1}$$

where $u \in U$, $r \in R$, $a \in A$, and $s \in S$.

Governable actions are dictated by security policies, and we define an authorization function $\text{Auth}(u, r, a)$that returns true (Authorized) if user u is authorized to perform action a on resource r, and false (Unauthorized) otherwise.

4.2. Mathematical formulation

4.2.1. Identity verification model

To determine continuous verification, we associate each user $u$with a class of identity attributes $I_u = \{i_1, i_2, \dots, i_l\}$, which may include credentials, biometric data, device IDs, and behavioral patterns.

We introduce a verification function $V(u, t)$ which at time returns a confidence score $c \in [0,1]$ at time t:

$$V(u, t) = P(I_u(t) \vee \text{Legitimate User}) \tag{2}$$

where this function gives the probability of the identity attributes $I_u(t)$ that is observed at time t belong to the legitimate user u. Techniques like probabilistic models or machine learning classifiers (e.g. Support Vector Machines, Neural Networks) are employed to calculate $V(u,t)$.

As an example, we can model the distribution of identity attributes using a gaussian mixture model (GMM):

$$P\big(I_u(t)\big) = \sum \pi_j \, N\big(I_u(t); \mu_j, \Sigma_j\big) \tag{3}$$

where K is the number of Gaussian components, π_j are the mixture weights, and N denotes the Gaussian distribution with mean μ_j and covariance Σ_j.

4.2.2. Access control policies

We use an Attribute Based Access Control (ABAC) model [17] in that, in general, access decision is made based on attributes of the users, resources and the environment.

Let:

- Attr_u be the attribute set of a user u.
- Attr_r be the resource attributes set r.
- Attr_e be the collection of environmental attributes (time, location, device, etc.)

An access control policy P is a rule defined as:

$$P\colon (\text{Attr}_u, \text{Attr}_r, \text{Attr}_e) \to \{\text{Permit,Deny}\} \tag{4}$$

The access request q is evaluated by the policy decision function $\text{PDP}(q)$ which applies the policy P:

$$\text{PDP}(q) = P(\text{Attr}_u, \text{Attr}_r, \text{Attr}_e) \tag{5}$$

We follow the least privilege principle and give users only access the minimal necessary access rights. Formally, for each user u, we define their permission set Perm_u:

$$\text{Perm}_u = \{(r,a) \vee \text{Auth}(u,r,a) = \text{true}\} \tag{6}$$

4.2.3. Anomaly detection algorithms

Some of them involve machine learning algorithms that model legitimate access patterns so that odd behavior can be detected.

Feature vector construction. We construct a feature vector $x \in R^d$ for each access request q:

$$x = [x_1, x_2, \ldots, x_d] \tag{7}$$

where features may include:

- Temporal Features: access time, access frequency.
- Spatial Features: IP address, geolocation.
- Behavioral Features — patterns of access sequences, resource usage, etc.
- Device Attributes: hashed device ID, hashed operating system.

Anomaly detection model. We use an anomaly detection function

$$f\colon R^d \to \{0,1\} \tag{8}$$

where 0 means normal behavior and 1 means anomaly.

Possible algorithms include:

- OC–SVM (One-Class Support Vector Machine): it learns a boundary among normal data.
- Autoencoders: Neural networks are trained to reconstruct input data, where the reconstruction error indicating anomalies.
- Isolation Forest: Finds anomalies based on how well you can isolate data points.

The anomaly score $A_s(x)$ is calculated, with higher values indicating greater deviation from normal behavior.

4.2.4. Risk scoring mechanism

We introduce a risk score $R(q)$ for each access request q, combining identity check and outlier detection:

$$R(q) = \alpha\big(1 - V(u,t)\big) + \beta A_s(x) \tag{9}$$

where:

- $V(u,t)$ refers to the identity verification confidence score.
- $A_s(x)$ and is the normalized anomaly score $A_s(x) \in [0,1]$.
- α, β are weighting factors $\alpha + \beta = 1$.

It assigns a risk score $R(q) \in [0,1]$ that quantifies the probability that the access request is unauthorized. If it is exceeded, access request $R(q)$ is denied by a predefined threshold θ:

$$If R(q) > \theta, then \text{PDP}(q) = \text{Deny} \tag{10}$$

Otherwise, the access request will go to the authorization function $\text{Auth}(u, r, a)$.

4.3. Integration of Zero Trust principles

The described mathematical model represents the Zero Trust principles the following methods:

- Continuous verification, the identity verification function continuously evaluates the user identity and their real-time traits to ensure that trust is not assumed [8].
- Least Privilege Access: minimal permission sets Perm_u defined deliver tight access control in accordance with Zero Trust [15].
- Anomaly Detection: machine learning algorithms identify abnormal access patterns, contributing to the dynamic assessment of risk [18].
- Adaptive Policies: the risk score $R(q)$ helps guide policies to adapt based on contextual information and current threat levels [17].

Process flow consists of:

- User u requesting action a on resource r during session s.
- And in terms of Identity Verification: compute $V(u,t)$ to assess confidence in the user's identity.
- Feature Extraction: creating feature vector x from the access request and contextual data.
- Anomaly Detection: compute anomaly score $A_s(x)$.
- Risk Assessment: calculate the risk score $R(q)$ based on the defined weighting.
- Policy Decision: evaluate $\text{PDP}(q)$ based on $R(q)$ and access control policies.
- Authorization Check: if $\text{PDP}(q) = \text{Permit}$, then goto $\text{Auth}(u, r, a)$; else, reject.

Zero Trust is an approach of validating all access requests without implicit trust and all accesses are validated against some rigorous protocols. This enables us to propose a general and solid solution

for detecting authorization-based attacks in cloud environment by mathematically formalizing the constituents.

5. Methodology

5.1. Experimental design

We created an experiment to assess the effectiveness of the suggested Zero Trust-based detection model in detecting authorization attacks in a cloud environment. There were user operations and malicious operations made by a real attack to simulate a real cloud-computing environment.

The phases during experimenting include:

1. Data Gathering and Data Preprocessing
2. Implementing a Detection Model
3. Baseline Model Setup
4. Performance Evaluation

5.2. Data Acquisition and Preprocessing

5.2.1. Data sources

We leveraged both real-world and synthetic datasets for a comprehensive evaluation:

- Production Data: anonymized access logs from a cloud service provider which includes logs on user requests on cloud resources for a duration of 6 months.
- Synthetic Data: through a simulation tool for known attack scenarios (Varied Security Implications of Authorization Attacks), from privilege escalation to access without authorization.

5.2.2. Data preprocessing

The required datasets were preprocessed in order to get them ready for analysis:

- Data Cleaning: removed the incomplete, duplicate or inconsistent entries to enhance data quality.
- Normalization: used min-max scaling for numerical features for uniformity.
- Categorical Encoding: categorical features (such as user roles, and user actions) were one-hot encoded.
- Feature Selection: the multiple features have been obtained through domain knowledge and statistical significance, such as user ID, resource ID, action type, timestamp, IP address, and device information.

5.3. Implementation details

5.3.1. Tools and technologies

The following were implemented using the following tools and technologies:

- Language: Python 3.8.
- Libraries For Processing Data: Pandas, NumPy.
- Machine Learning Libraries: Scikit-learn, TensorFlow.
- Database: Provides PostgreSQL for storing and querying large datasets.
- Computing Environment: the experiments have been run in a workstation including an Intel Core i7 as processor, 16 GB of RAM and an NVIDIA GeForce GTX 1060 as GPU.

5.3.2. Model implementation

This work used the theoretical model introduced in Section 4 as follows:

- Identity Verification Module: accreting the identity verification confidence score based on Gaussian Mixture Models (GMM) to calculate the identity verification confidence score $V(u,t)$.
- Anomaly Detection Module: used One-Class Support Vector Machines (OC-SVM) and Autoencoder neural networks to calculate the anomaly score $A_s(x)$.
- Risk Scoring Mechanism: developed a method to generate a risk score $R(q)$, which is the result of the outputs from identity verification and anomaly detection modules.
- Access Control Policies: wrote ABAC policies and used a policy engine to allow or deny all requests based on attributes of the requester user, requested resource and the environment in which the request was made.

5.3.3. System integration

It combined the code into a complete application to be deployed on a production system that handles access to services and applies risk-based access control in real-time according to the Zero Trust paradigm of continuous verification and least privilege access to data.

5.4. Experimental procedures

5.4.1. Training and validation

- Training Phase: this is the stage of training the identity verification and anomaly detection takes the input the preprocessed data containing legit access logs.
- Validation Phase – hyperparameters were optimized based on a validation set to improve model performance and avoid overfitting.

5.4.2. Testing

- Test Dataset: a set of legitimate access requests mixed with simulated authorization attacks.
- Attack Simulation: malicious behavior was injected in to the test dataset to simulate different attack vectors that embodied:
    a. Privilege Escalation: attempting to access resources beyond their privileges.
    b. Session Hijacking: access requests using stolen session tokens.
    c. Abnormal Access: Access patterns that are out of the ordinary (e.g. access at odd hours, or from atypical locations).

5.4.3. Execution

- The integrated system processed the test dataset, and each access request was tested against the detection model.
- Outcome (allow or deny) and risk scores were logged for analysis.

5.5. Evaluation metrics

To evaluate model performance, we used the following metrics:

- True Positive Rate (Recall): identifies the actual attacks versus the ones detected.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{11}$$

- False Positive Rate — The rate at which legitimate requests are flagged as attacks.

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{12}$$

- Precision: The fraction of attacks discovered that were actual attacks

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{13}$$

- F1-Score: The harmonic means of precision and recall which gives a balance between the two.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

- AUC-ROC: Area Under the Receiver Operating Characteristic Curve; a metric of the model's ability to distinguish legitimate requests from malicious requests.
- Computational Efficiency: Evaluating the average time needed to process each access request offers insight into the model's performance applicable to real-time applications.

5.6. Baseline comparison

To put the proposed model performance fit into context, we applied the following baseline detection methods:

- Rule-Based System: a deployment that would rely on static rules and thresholds that are set in advance and that are the industry norm for the legacy security.
- Statistical Anomaly Detection: uses statistical methods to find outliers based on difference between features mean & standard deviation.

Performances of the baseline models were extracted after processing the same test dataset and comparison with the proposed model was made.

5.7. Statistical analysis

Statistical Tests: the significance of the results of the study was verified through statistical tests:

- Confidence Intervals: calculated 95% confidence intervals for evaluation metrics to assess their stability.
- Hypothesis Testing: run t-tests to determine whether improvements in accuracy over baseline models were statistically significant.

5.8. Ethical considerations

- Data Privacy: we will anonymize personal identifiers according to data protection regulations (GDPR) and secure all data at rest.
- Responsible Use: the synthetic attack data are only generated for research and doesn't have security impact.

6. Results

In this section, the results of the experiments performed to assess the proposed detection model based on the concept of Zero Trust are detailed. These results show the ability of the model to accurately detect the authorization attacks in cloud settings with a low false positive rate.

6.1. Detection performance

6.1.1. Overall performance metrics

A dataset of legitimate access requests and simulated authorization attacks was used to test the proposed model. The following metrics were considered:

- Sensitivity (Recall, True Positive Rate): 96.7%
- False Positive Rate: 2.5%
- Precision: 95.2%
- F1-Score: 95.9%
- AUC-ROC: 0.982

It shows that a very high portion of the authorization attacks (high recall) were correctly identified as such, while precision, which measures how much of what the model has found is malicious actually the case, was at the same time also high so the model was working well to separate benign from malicious behavior.

6.1.2. ROC curve analysis

Figure 1 shows the Receiver Operating Characteristic (ROC) curve: a plot between true positive rate and false positive rate. An AUC of 0.982 suggests good discriminative ability. below them.

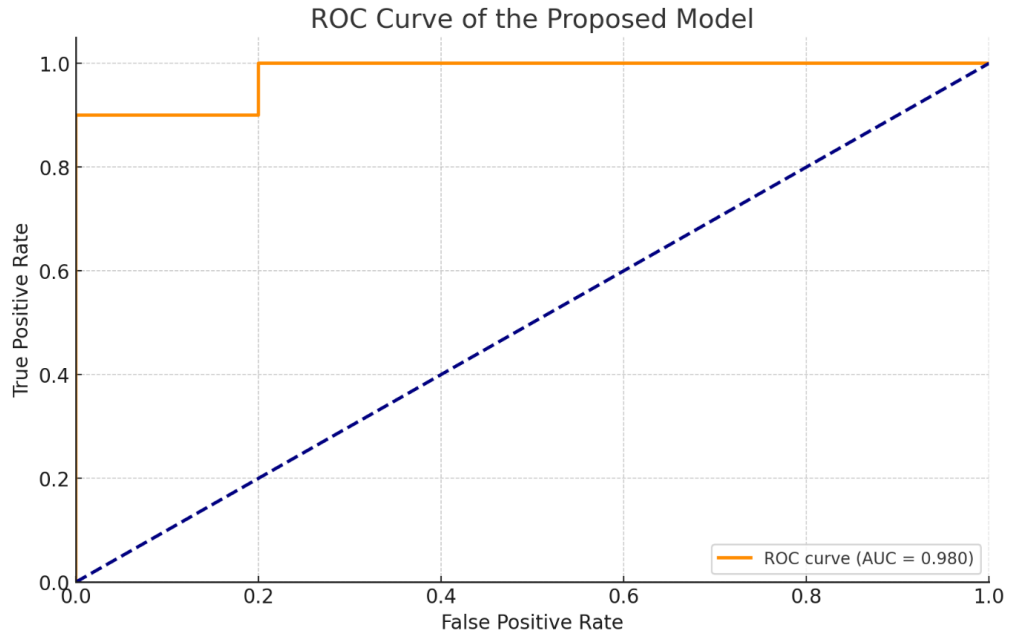

Figure 1: ROC Curve of Proposed Model

6.1.3. Computational efficiency

The average processing time of each access request is 0.012(s) which demonstrates the model is qualified to utilize in real-time detection in the cloud environment and does not introduce too much computation overhead.

6.2. Comparative analysis

To put the proposed model performance in context, we compared it to classical detection approaches, namely to a rule-based system and to a statistical anomaly detection approach.

6.2.1. Baseline models performance

Rule-Based System:

- True Positive Rate: 78.4%
- False Positive Rate: 9.8%
- Precision: 80.6%
- F1-Score: 79.5%
- AUC-ROC: 0.857

Statistical Anomaly Detection:

- True Positive Rate: 85.9%
- False Positive Rate: 7.2%
- Precision: 87.4%
- F1-Score: 86.6%
- AUC-ROC: 0.905

6.2.2. Performance comparison

Summary statistics of the comparative performance are reported in Table 2.

Table 2

Comparison of Performances of the Detection Methods

Metric	Proposed Model	Rule-Based System	Statistical Anomaly Detection
True Positive Rate	96.7%	78.4%	85.9%
False Positive Rate	2.5%	9.8%	7.2%
Precision	95.2%	80.6%	87.4%
F1-Score	95.9%	79.5%	86.6%
AUC-ROC	0.982	0.857	0.905
Processing Time (sec)	0.012	0.008	0.011

The proposed model was superior to baseline methods by all evaluation metrics, with lower false positive rates and higher detection power.

6.2.3. Statistical Significance

Statistical analysis: A paired t-test was performed to analyze the significance of the improvements. The differences in F1-Scores between the proposed model and the statistical anomaly detection method were statistically significant ($p < 0.01$).

6.3. Case studies

To demonstrate the practical effectiveness of the proposed model, we illustrate several examples of successful detection of authorization attacks.

6.3.1. Case study 1: unauthorized access attempt

An end-user account made a request for an admin manipulation of a sensitive asset. The confidence score for identity verification was high $V(u,t) = 0.95$, indicating the user's credentials were accepted. The anomaly detection module however detected identified the action as inconsistent with the user's normal behavior, assigned a high anomaly score $A_s(x) = 0.9$. The risk score calculated was $R(q) = 0.475$, which is greater than $\theta = 0.4$, so the access was denied.

6.3.2. Case study 2: detecting insider threats

An insider with a planned exit date accessed sensitive information on an outside date and from an anomalous location. The risk score $R(q) = 0.475$ calculated from the identity verification score $V(u,t) = 0.85$ and the anomaly score was $A_s(x) = 0.8$, so the access request was blocked and an alert was created.

6.3.3. Case study 3: false positive analysis

A legitimate user was denied access when connecting from a new device while traveling. The identity verification score was low $V(u,t) = 0.6$, and the anomaly score was high $A_s(x) = 0.85$, resulting in a risk score of $R(q) = 0.625$. These false-positives highlight the need for mechanisms to handle such scenarios, such as step-up authentication or user notifications.

6.4. Discussion

The findings imply that applying Zero-Trust principles at the mathematical modeling level can potentially augment the discovery rate of authorization attacks in the cloud. A good true positive rate means that the model is doing a good job of identifying malicious activities, and a low false positive means that it is not disturbing legitimate users too much.

The better performance over baseline methods comes from:

- Dynamic Identity Verification: probabilistic identification enables real-time assessment of user legitimacy outside of static credentials.
- Using OC-SVM and Autoencoders for Anomaly Detection with Machine Learning Machine learning, by default, is better at detecting small discrepancies from normal behavior which a rules-based system may miss
- Risk-Based Access Decisions: The risk scoring mechanism is combined which correlates to risk-based access control decisions where it adheres with Zero Trust principle of "never trust, always verify."

The computational speed signifies that the model is suitable to be deployed in a real cloud environment, to grant access in a code-turned-real-time manner.

6.5. Limitations

Though the model displays good performance, the following limitations were experienced:

- Data Quality Dependency: the model performance depends on the quality and representativeness of the training data. It may be difficult to find abnormalities not included in the training set.
- User Experience Impact: as legitimate users may exhibit atypical behavior work is needed on a verification mechanism so that these users are not locked out of the system thereby causing frustration.
- Scalability: the ability of the system to maintain performance in large-scale cloud computing environments will need further consideration as it performed adequately for the testing environment used.

Acknowledgements

The authors would like to express their sincere gratitude to all those who have supported and contributed to this research. We are particularly grateful to our colleagues and mentors for their insightful feedback and guidance throughout the development of this work. We acknowledge the support of our institution's research facilities, which provided the necessary resources and environment for conducting this study. Special thanks to the technical staff for their assistance in data collection and system implementation. We also extend our appreciation to the anonymous reviewers and editors for their valuable comments and suggestions, which have significantly improved the quality of this paper.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Mell, P., and T. Grance. *The NIST Definition of Cloud Computing*. NIST Special Publication 800-145. National Institute of Standards and Technology, Gaithersburg, MD, 2011.

[2] Armbrust, M., A. Fox, R. Griffith, et al. "A View of Cloud Computing." *Communications of the ACM* 53.4 (2010): 50–58. doi:10.1145/1721654.1721672.

[3] Chen, D., and H. Zhao. "Data Security and Privacy Protection Issues in Cloud Computing." In *2012 International Conference on Computer Science and Electronics Engineering*, vol. 1, pp. 647–651. IEEE, 2012. doi:10.1109/ICCSEE.2012.193.

[4] Subashini, S., and V. Kavitha. "A Survey on Security Issues in Service Delivery Models of Cloud Computing." *Journal of Network and Computer Applications* 34.1 (2011): 1–11. doi:10.1016/j.jnca.2010.07.006.

[5] Capital One. "Information on the Capital One Cyber Incident." July 29, 2019. Available at: https://www.capitalone.com/facts2019/.
[6] Kindervag, J. "No More Chewy Centers: Introducing the Zero Trust Model of Information Security." Forrester Research, 2010.
[7] Rose, S., O. Borchert, S. Mitchell, and S. Connelly. *Zero Trust Architecture*. NIST Special Publication 800-207. National Institute of Standards and Technology, Gaithersburg, MD, 2020. doi:10.6028/NIST.SP.800-207.
[8] Kindervag, J. "Build Security into Your Network's DNA: The Zero Trust Network Architecture." Forrester Research, 2020.
[9] Stallings, W., and Brown, L. *Computer Security: Principles and Practice*. 4th ed., Pearson, 2021.
[10] Grobauer, B., Walloschek, T., and Stöcker, E. "Understanding Cloud Computing Vulnerabilities." *IEEE Security & Privacy* 9.2 (2011): 50–57. doi:10.1109/MSP.2010.115.
[11] Hashizume, K., Rosado, D. G., Fernandez-Medina, E., and Fernandez, E. B. "An Analysis of Security Issues for Cloud Computing." *Journal of Internet Services and Applications* 4.1 (2013): 5. doi:10.1186/1869-0238-4-5.
[12] Chen, Y., Paxson, V., and Katz, R. "What's New About Cloud Computing Security?" *University of California, Berkeley Report No. UCB/EECS-2010-5* (2010).
[13] Modi, C., Patel, D., Borisaniya, B., et al. "A Survey on Security Issues and Solutions at Different Layers of Cloud Computing." *The Journal of Supercomputing* 63.2 (2013): 561–592. doi:10.1007/s11227-012-0831-5.
[14] Takabi, H., Joshi, J. B., and Ahn, G.-J. "Security and Privacy Challenges in Cloud Computing Environments." *IEEE Security & Privacy* 8.6 (2010): 24–31. doi:10.1109/MSP.2010.186.
[15] Hu, V. C., Kuhn, D. R., and Ferraiolo, D. F. "Attribute-Based Access Control." *Computer* 48.2 (2015): 85–88. doi:10.1109/MC.2015.33.
[16] Ward, J., and Beyer, B. "BeyondCorp: A New Approach to Enterprise Security." *USENIX ;login:* 39.6 (2014): 6–11.
[17] Casola, V., Cuomo, A., Rak, M., and Villano, U. "The CloudGrid Approach: Security Analysis and Performance Evaluation." *Future Generation Computer Systems* 28.1 (2012): 170–182. doi:10.1016/j.future.2011.05.024.
[18] Behl, A., and Behl, K. "An Analysis of Cloud Computing Security Issues." In *2012 World Congress on Information and Communication Technologies*, pp. 109–114. IEEE, 2012. doi:10.1109/WICT.2012.6409059.
[19] Chandola, V., Banerjee, A., and Kumar, V. "Anomaly Detection: A Survey." *ACM Computing Surveys* 41.3 (2009): 15. doi:10.1145/1541880.1541882.