Ms.Dhivya K, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science,
Engineering and Technology
An Open Access Journal

TerraSecure: A Machine Learning Framework for Detecting Infrastructure as Code Misconfigurations with 10.7% False Positive Rate

Ms.Dhivya K, Bhavayazhinitha S V, Gunal S, Jashwanth M U, Kanishka R, Gokulnath K
Department of Computer Science & Engineering (Cyber Security) Sri Shakthi Institute of Engineering and Technology Coimbatore, Tamilnadu, India.

Abstract: However, the misconfiguration in the IaC template is now regarded as one of the critical factors responsible for cloud security breaches. Services impacted by these include storage, networking, identity management, and databases. This paper discusses TerraSecure – an advanced intelligent multilayer framework that is capable of identifying such misconfigurations. TerraSecure applies a hybrid approach which includes rule-based detection, machine learning, and AI-powered contextual analysis. This framework employs more than 50 security patterns extracted from actual breaches as well as best practices in cloud computing. A pre-trained XGBoost model, considering 50 security patterns, predicts the risk score with accuracy of 92.45% while ensuring the minimal false-positive ratio of 10.71%. As a result, vulnerable configurations, such as public storage access, overly broad permission scopes, unencrypted data, and unsafe network settings, can be identified. Moreover, an AI analysis component adds to the interpretability of this framework by delivering information about potential business impact, attack scenario (based on the real incident), and remediation steps. In addition, TerraSecure supports several output formats, among which there is SARIF to facilitate the integration with CI/CD pipelines and other tools (e.g., GitHub). The conducted experiments confirm the scalability, efficiency, and reliability of this framework in terms of security

Keywords: Infrastructure as Code (IaC), Cloud Security, Misconfiguration Detection, Machine Learning, XGBoost, Artificial Intelligence, DevSecOps, Risk Assessment

I. INTRODUCTION

The fast pace of cloud computing implementation has changed the approach to designing and deploying IT infrastructure. One of the factors that have enabled this change is Infrastructure as Code (IaC). IaC makes it possible for developers to deploy and maintain cloud infrastructure using machine-readable code. The benefits of using IaC include automation, scalability, consistency, and accelerated deployment. Therefore, it has become a crucial element of modern DevOps and DevSecOps practices.

However, while being useful, IaC brings its own set of risks. For instance, misconfiguration in infrastructure code is now one of the most common reasons for cloud security incidents. Such errors can be related to exposing storage to the public Internet, granting excessive permissions to users, storing sensitive information without encryption, and other issues. These types of problems can cause data breaches, financial losses, and regulatory penalties. In addition, current security solutions are focused on finding vulnerabilities in code by using predetermined rules. However, these solutions cannot detect all types of vulnerabilities, and their use is prone to generating numerous false positives.

In an attempt to overcome these shortcomings, this paper recommends a framework called TerraSecure

that makes use of intelligent techniques such as machine learning and artificial intelligence in order to detect and analyze misconfiguration issues in cloud computing. TerraSecure incorporates rule-based detection techniques along with machine learning algorithms in order to achieve more effective results. Specifically, TerraSecure makes use of a pre-built XGBoost machine learning algorithm in order to determine the risk scores using various security attributes. Furthermore, an AI layer analyzes the results and provides insights about business impacts, attack vectors, and remediation steps.

TerraSecure uses intelligent prediction and reasoning in order to achieve more effective results in terms of cloud security. The proposed methodology can be easily integrated into CI/CD pipelines.

II. LITERATURE REVIEW

As more cloud-native technologies emerge, different strategies were devised for addressing the problem of security threats within IaC. For example, Checkov, developed by Bridgecrew, offers static analysis of IaC templates to find misconfigurations based on preconfigured rules [6]. Likewise, Trivy, created by Aqua Security, allows vulnerability scanning for containers and IaC configurations [7]. Nevertheless, these tools are primarily based on rules that may lead to false positives and miss contextual risk analysis.

The recent literature addresses the issue of misconfiguration detection. The PHOENIX framework by J. Wen and H. Ping is an example of automation of the process of detecting vulnerabilities within AWS serverless environments [1]. Furthermore, M. A. Ghorbani and M. A. Saied discuss the problems associated with Kubernetes configuration and security [2]. This emphasizes the importance of developing advanced and context-aware techniques.

HashiCorp's guidelines on implementing Terraform security best practices [3], as well as the NIST National Vulnerability Database (NVD) [4] and MITRE Common Vulnerabilities and Exposures (CVE) [5], contain basic information on finding misconfigurations and security flaws. Moreover, large-scale studies indicate the ubiquity of vulnerabilities in cloud deployments [8].
There have been excellent results in predictive analysis due to the use of machine learning methods, such as the XGBoost algorithm [9]. There is also an opportunity to utilize AI through platforms such as AWS Bedrock to provide contextually insightful data [10].

However, there is no solution that can integrate all these methods. This is where the contribution of TerraSecure comes in.

III. FRAMEWORK OF THE SYSTEM

TerraSecure is a layered approach which detects, analyzes and corrects cloud misconfigurations for IaC. It utilizes rule-based detection, machine learning and artificial intelligence to achieve highly accurate cloud security assessments.

1. Input Layer

This layer includes IaC files like Terraform configurations, modules, and HCL scripts, which act as the inputs for further security analysis.

2. Parsing & Feature Extraction Layer

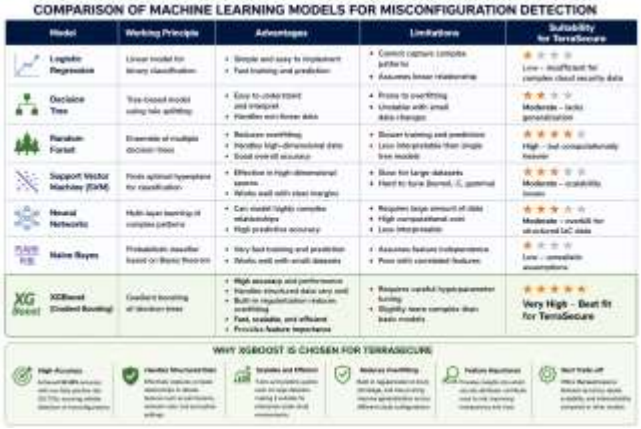
At this stage, IaC files are parsed with an HCL parser. Relevant resources and their attributes (e.g., S3 buckets, security groups, IAM roles, etc.) are extracted.

3. Detection Engine

Detection engine contains the following elements:
Rule-based analyzer: It performs over 50 patterns that are derived from actual cloud breaches. These patterns ensure identification of all known misconfigurations.
Feature extractor: It generates 50 features from the extracted resources.

4. Machine Learning Layer

Table 1: Comparison of ML Models



The input provided to pre-trained XGBoost machine learning model is the security features from previous layers. Output is the prediction of risk score (0 to 1) and confidence value per each feature.

5. AI Analysis Layer

This layer utilizes AI methods to further improve the detected findings. The following outputs are provided:

- Impact of the finding on business operations
- Attack scenario based on historical data
- Detailed actions to fix the issue
- Intelligent backup analysis system that works independently of third-party AI providers.

6. Output Layer

Output is available in different formats including Text, JSON, and SARIF formats.

7. Integration Layer

TerraSecure is capable of working in CI/CD pipelines seamlessly (e.g., GitHub Actions).

IV. WORKFLOW OF TERRASECURE

In the process described in TerraSecure documentation, the first step includes collecting IaC files, such as terraform configurations, modules, and HCL scripts. Next, these input files are parsed by HCL parser that extracts required resources together with their attributes. After extracting these resources, they can be analyzed with a rule-based detection mechanism that uses more than 50 security rules. These rules are created based on previous cloud breaches and best practices within the industry. Some examples of these security rules include checking whether there is publicly accessible storage bucket, IAM policies that are too permissive, unencrypted data storage, and vulnerable network configurations.

Figure 1: Workflow of Terrasecure

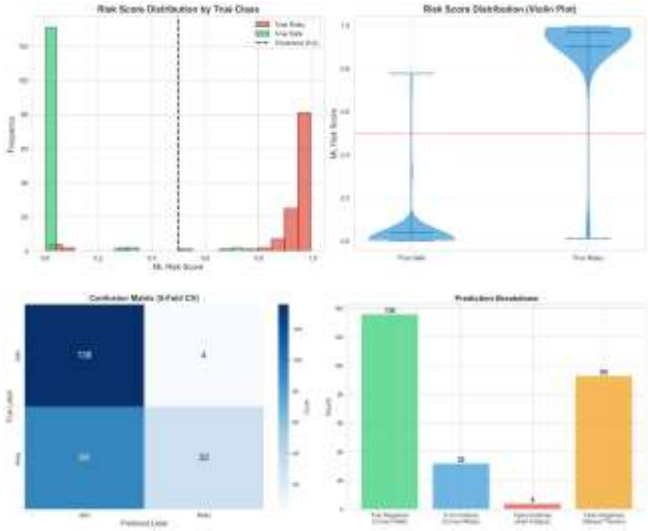
After performing rule-based detection, it is time to extract security features from the resources detected during the previous step. In total, TerraSecure generates around 50 security features for each resource detected by the rule-based mechanism. Such features include access control options, encryption, public accessibility, and monitoring configurations. Finally, once all features are collected for each resource, they are provided to the pre-trained XGBoost machine learning algorithm that computes the risk score between 0 and 1 along with its confidence level.

In addition to improving the usability of the analysis results, TerraSecure includes an additional smart AI-based analysis stage. The output of this stage contains all necessary context information, including a description of any problems found, business impact estimation, historical examples of similar attacks from real cases, and instructions for fixing the problem. A special failure-resistant system is implemented that will make sure the analysis is not interrupted in case of problems with external AI systems.

Finally, the results of the analysis can be output in several formats, including human-readable text, machine-readable JSON, and SARIF. All these formats will enable integration with various CI/CD systems, such as GitHub Actions, where security checks can be automated.

V. RESULTS AND DISCUSSION

The assessment of the TerraSecure system shows its efficiency in detecting and evaluating security misconfigurations in IaC environments. The TerraSecure system was evaluated using various datasets, which were obtained from actual cloud infrastructures and attacks against them. These datasets contained misconfigurations of a variety of services, including storage, networking, identity management, and databases. According to the evaluation results, the machine learning model implemented within TerraSecure using the XGBoost algorithm achieved 92.45% accuracy with only 10.71% of false positives.



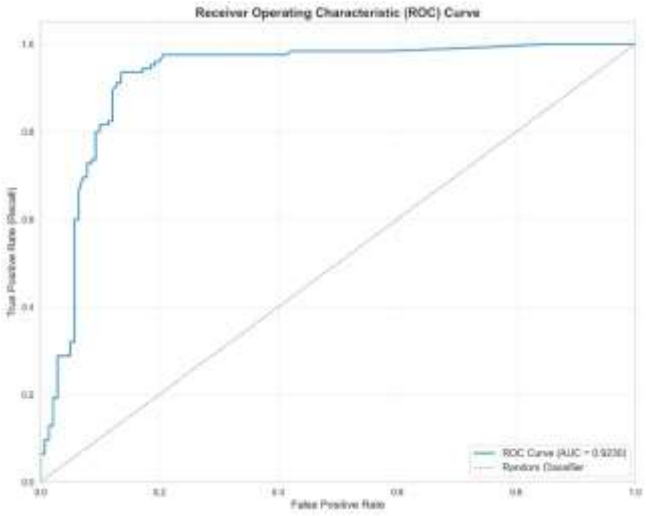


Figure 2: Risk Score Distribution, Confusion Matrix and Prediction Breakdown, ROC Curve

Combining the rule-based detection engine with the machine learning model resulted in increased detection rates. Although the rule-based detection engine effectively detected known misconfigurations, the machine learning engine improved the detection rates for unknown misconfigurations and allowed for their detection. Thus, combining rule-based detection and machine learning overcomes the disadvantages of existing static analysis tools because the former rely solely on predefined rules.

Moreover, incorporating the AI-based analysis layer made the process more understandable and user-friendly. Offering additional context information like business impacts, practical attack examples, and steps to remediate identified issues, TerraSecure ensures that security experts can base their actions on sound reasoning and correctly prioritize potential vulnerabilities. Intelligent fallback behavior made TerraSecure perform reliably even when external AI tools were unavailable.

Speaking about its performance characteristics, TerraSecure showed high scalability and efficiency – TerraSecure managed to scan massive infrastructure setups in a relatively short period of time with minimal

resource usage. It is noteworthy that TerraSecure could produce results in formats like SARIF which allowed it to be easily integrated into CI/CD processes.
Thus, taking all things into consideration, one may conclude that TerraSecure provides a reliable and highly efficient tool for analyzing cloud misconfigurations, demonstrating better results than other similar solutions currently available on the market.

VI. CONCLUSION

Thus, in the context of this research paper, TerraSecure is considered an innovative and advanced solution for detecting and resolving misconfiguration issues in the IaC-driven environment. The introduction of the rule-based analysis approach, machine learning algorithms, and artificial intelligence helps to eliminate the weaknesses of traditional approaches that employ the application of static rules only. It becomes possible to conduct risk assessment in a highly accurate manner due to the use of the pre-trained XGBoost classifier. Meanwhile, the presence of the AI module will help improve the understanding of the received outcomes since it provides insights into the possible business impact, threat vectors, and needed actions.

On the basis of the obtained results, TerraSecure is capable of providing highly efficient and reliable security assessments. Its capacity to combine predictive models and security patterns allows for detecting new threats effectively. Furthermore, the compatibility with different outputs and CI/CD pipelines contributes to the practicality of TerraSecure.
TerraSecure is an innovative method that enhances the security of clouds by helping proactively discover and fix vulnerabilities. Potential areas for future research would involve expanding the functionality to incorporate multiple clouds, implementing more machine learning models, and improving real-time analysis performance.

Future Work
In spite of high efficiency and effectiveness of the framework at finding and analyzing configuration issues within cloud environment, there is still a wide range of approaches to making TerraSecure even more efficient and more powerful when dealing with more complicated cases. First of all, one should develop support for such cloud platforms as Microsoft Azure and Google Cloud Platform, which would allow expanding the scope of application of the framework and working with more various types of clouds. In addition, the authors can introduce more code formats for infrastructure as code such as Kubernetes manifests and ARM templates.

There is also a number of improvements associated with the integration of advanced machine learning and deep learning approaches that will be useful for improving predictions made by the system and reducing the number of errors detected. In particular, continuous training on new data will enhance the adaptability of the system.

REFERENCES

1. J. Wen and H. Ping, "PHOENIX: Misconfiguration Detection for AWS Serverless Computing," IEEE Transactions on Cloud Computing, vol. 13, no. 3, pp. 922–934, Jul.–Sep. 2025.
2. M. A. Ghorbani and M. A. Saied, "Towards Secure Cloud-Native Computing: Unveiling Kubernetes Misconfigurations," IEEE Access, 2024.
3. "Terraform Best Practices for Security," HashiCorp Documentation, 2023.
4. National Institute of Standards and Technology (NIST), "National Vulnerability Database (NVD)," 2024.
5. MITRE, "Common Vulnerabilities and Exposures (CVE) Database," 2024.
6. Bridgecrew, "Checkov: Static Code Analysis for Infrastructure as Code," 2023.
7. Aqua Security, "Trivy: Comprehensive Security Scanner," 2023.

8. J. Chen et al., "A Large-Scale Study on Security Vulnerabilities in Cloud Deployments," IEEE Cloud Computing, 2022.
9. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD, 2016.
10. Amazon Web Services, "AWS Bedrock: Foundation Models for Generative AI," 2024.