**SECURON: AN INTELLIGENT FRAMEWORK FOR REALTIME CLOUD THREAT DETECTION AND AUTOMATED SECURITY ENFORCEMENT**

*Dr. P. Velmurugadass*

*Associate Professor Department of Computer Science and Engineering Kalasalingam Academy of Research and Education Krishnankoil, 626126, India* [*velmurugadass.p@gmail.com*](mailto:velmurugadass.p@gmail.com)

*Barathi M*

*Department of Computer Science and Engineering Kalasalingam Academy of Research and Education Krishnankoil, 626126, India*

[*99220041124@klu.ac.in*](mailto:99220041124@klu.ac.in)

*Remina S*

*Department of Computer Science and Engineering Kalasalingam Academy of Research and Education Krishnankoil, 626126, India*

[*99220040712@klu.ac.in*](mailto:99220040712@klu.ac.in)

*Nivetha M*

*Department of Computer Science and Engineering Kalasalingam Academy of Research and Education Krishnankoil, 626126, India*

[*99220040666@klu.ac.in*](mailto:99220040666@klu.ac.in)

*Meghana N*

*Department of Computer Science and Engineering Kalasalingam Academy of Research and Education Krishnankoil, 626126, India*

[*99220040660@klu.ac.in*](mailto:99220040660@klu.ac.in)

***Abstract- Cloud security failures frequently arise from misconfigured Infrastructure-as-Code (IaC) and insufficient detection of runtime threats. Existing solutions typically focus on either static analysis before deployment or monitoring after deployment, leaving critical security gaps. This paper introduces Securon, a closed-loop cloud security framework that integrates Terraform-based static misconfiguration detection with unsupervised machine learning analysis of AWS VPC Flow Logs and CloudTrail events. Securon identifies security risks both before deployment and during runtime, including port scans, brute-force attacks, and privilege misuse. The system generates adaptive security rules from detected anomalies, which are validated through a human-in-the-loop dashboard and reintegrated into the IaC scanning engine. Experiments using synthetic and semi-realistic logs demonstrate effective threat detection with minimal false positives, highlighting Securon’s potential as an adaptive DevSecOps security solution.***

***Keywords - Cloud Security, Infrastructure-as-Code (IaC), Terraform, Machine Learning, Anomaly Detection, AWS VPC Flow Logs, CloudTrail, DevSecOps, Static Analysis, Runtime Monitoring***

I. INTRODUCTION

*A. Background and Motivation*

Cloud computing allows for quick infrastructure provisioning and scalability. However, it also brings new security challenges. The rise of Infrastructure-as-Code (IaC) and CI/CD pipelines has increased the risk of security misconfigurations. These can include publicly exposed services, overly permissive IAM roles, and unencrypted storage resources. Industry reports show that most cloud security incidents come from configuration errors rather than software vulnerabilities [8], [24].

*B. Limitations of Existing Security Approaches*

Most traditional cloud security tools detect issues after deployment. Runtime security platforms like AWS GuardDuty and Splunk only identify suspicious behavior once the infrastructure is already exposed [2], [17]. Likewise, static IaC scanners can prevent misconfigurations before deployment but do not consider runtime behavior, allowing new and context-specific threats to go undetected.

*C. Contribution of This Work*

This paper introduces Securon, a unified cloud security framework that links pre-deployment and post-deployment security. By combining Terraform-based static analysis with unsupervised machine learning on AWS VPC Flow Logs and CloudTrail events, Securon creates a closed-loop security system that continuously improves IaC policies using runtime intelligence.

II. LITERATURE REVIEW

*A. Cloud Security Posture Management (CSPM)*

Tools CSPM platforms like Prisma Cloud, Wiz, and Lacework focus on finding misconfigurations in deployed cloud environments [9], [19]. While they are effective for compliance and visibility, these tools work post-deployment and do not integrate directly with IaC pipelines.

*B. Static IaC Analysis Tools*

Tools such as Checkov, tfsec, and Terrascan analyze Terraform templates for known misconfigurations [5]–[7]. These tools depend on manually curated rule sets and cannot adjust to new threat patterns or runtime behaviors.

*C. Runtime Monitoring and Log Analysis*

Runtime detection systems, including AWS GuardDuty, ELK Stack, and Splunk, analyze cloud telemetry like VPC Flow Logs and CloudTrail events to spot malicious activities [2], [16], [17]. However, these systems are reactive and do not impact future infrastructure configurations.

*D. Machine Learning for Cloud Anomaly Detection*

Unsupervised machine learning methods such as Isolation Forest, autoencoders, and LSTM models have shown strong results in spotting anomalies in cloud logs [12], [13], [15], [20]. However, existing methods do not integrate with IaC enforcement pipelines, limiting their long-term preventive value.

III. PROPOSED METHODOLOGY

*A. Pre-Deployment IaC Validation*

Securon starts by reviewing Terraform execution plans created during CI/CD workflows. These plans show complete infrastructure details, including network rules, IAM permissions, and encryption settings. The static scanning engine checks these details against a set of security rules to stop insecure configurations before deployment [4].

*B. Runtime Log Collection and Preprocessing*

After deployment, AWS VPC Flow Logs and CloudTrail logs provide data on network and identity activity. The raw logs are preprocessed by normalizing timestamps, encoding categories, adding geographic information to IP addresses, and calculating metrics like port diversity and request frequency.

*C. Anomaly Detection using Machine Learning*

Securon uses the Isolation Forest algorithm for detecting anomalies without supervision. This model identifies abnormal behavior without needing labeled attack data, making it effective for large-scale cloud activity [12]. Notable anomalies include port scans, brute-force attempts, unusual API use, and abnormal data transfers.

*D. Human-in-the-Loop Rule Reinforcement*

Detected anomalies turn into candidate security rules. These rules are shown to security analysts on the dashboard, where they can approve or reject them. Approved rules are reintegrated into the static IaC scanner, allowing for flexible policy enforcement.

IV. ARCHITECTURE AND WORKFLOW

*A. High-Level System Architecture*

* Securon has four main components:
* Terraform engine
* Static IaC scanning engine
* Runtime log analysis and ML engine
* Dashboard and enforcement layer

*B. End-to-End Workflow*

The workflow starts with generating a Terraform plan, followed by static security validation. After deployment, runtime logs go into the ML engine for detecting anomalies. Detected anomalies create candidate rules, which are reviewed and enforced through the dashboard.

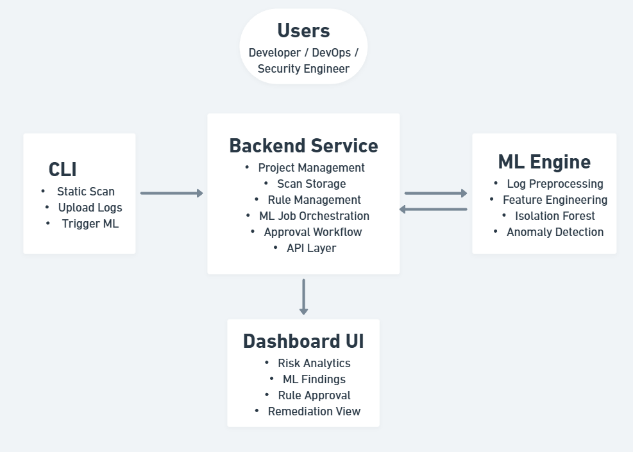


Fig. 1. End-to-end data flow architecture of Securon.

This figure illustrates how data moves from IaC configurations to runtime logs, ML analysis, and back into the IaC enforcement pipeline.

*C. Closed-Loop Security Model*

Securon uses a closed-loop security model where runtime findings continuously improve pre-deployment validation. This approach prevents recurring vulnerabilities and strengthens security over time.

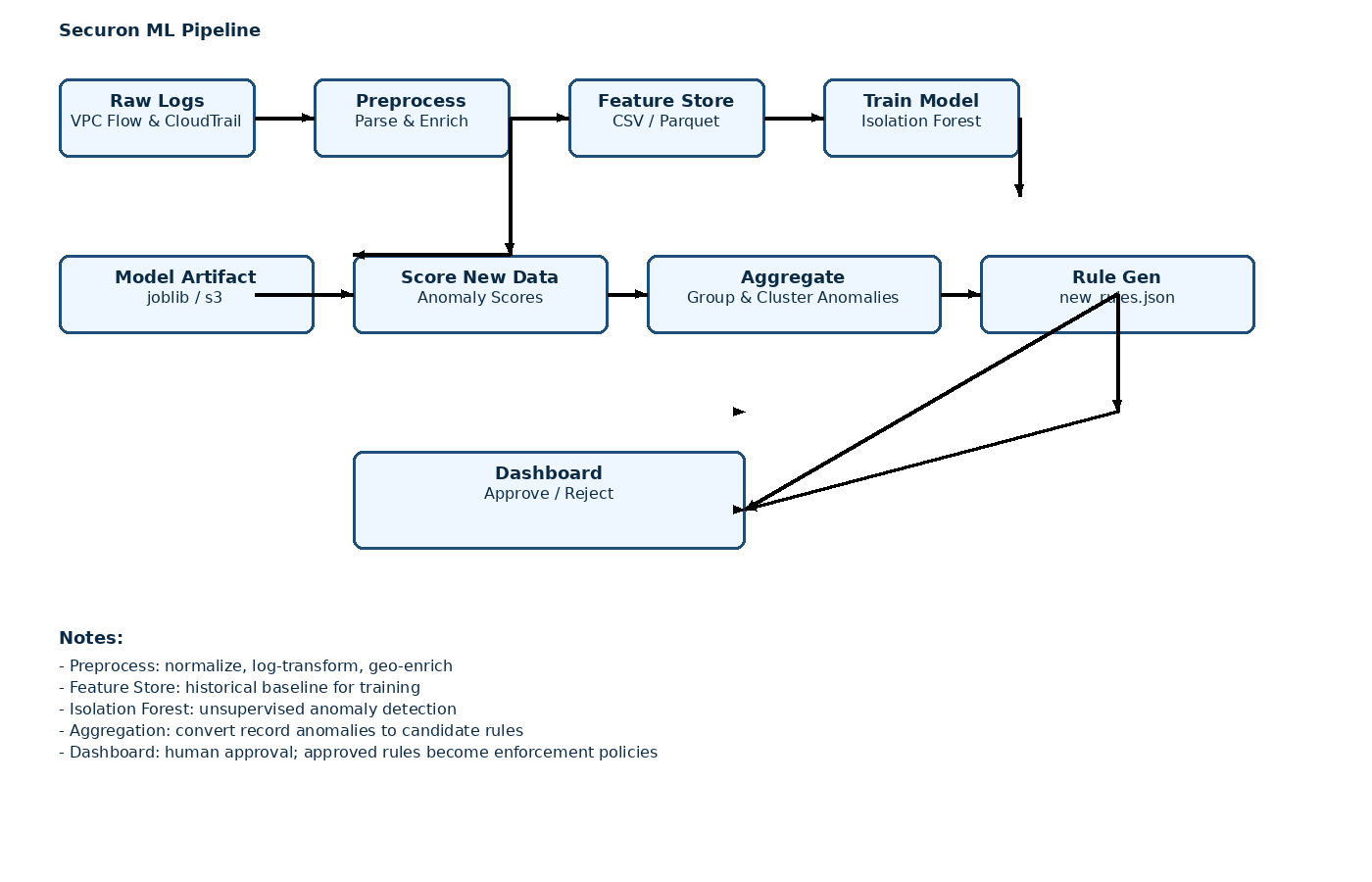


Fig. 2. Closed-loop security model showing continuous policy reinforcement.

This figure emphasizes the feedback loop between runtime detection and the improvement of static IaC rules.

V. RESULTS AND DISCUSSION

The evaluation of Securon shows its effectiveness in three areas: detecting IaC misconfigurations, identifying runtime anomalies, and generating adaptive security rules. The following outputs confirm that each system component functions correctly.

*A. Static IaC Misconfiguration Detection*

Securon flagged Terraform misconfigurations such as unrestricted ingress rules, publicly accessible databases, and unencrypted storage. These issues were prevented from being deployed.

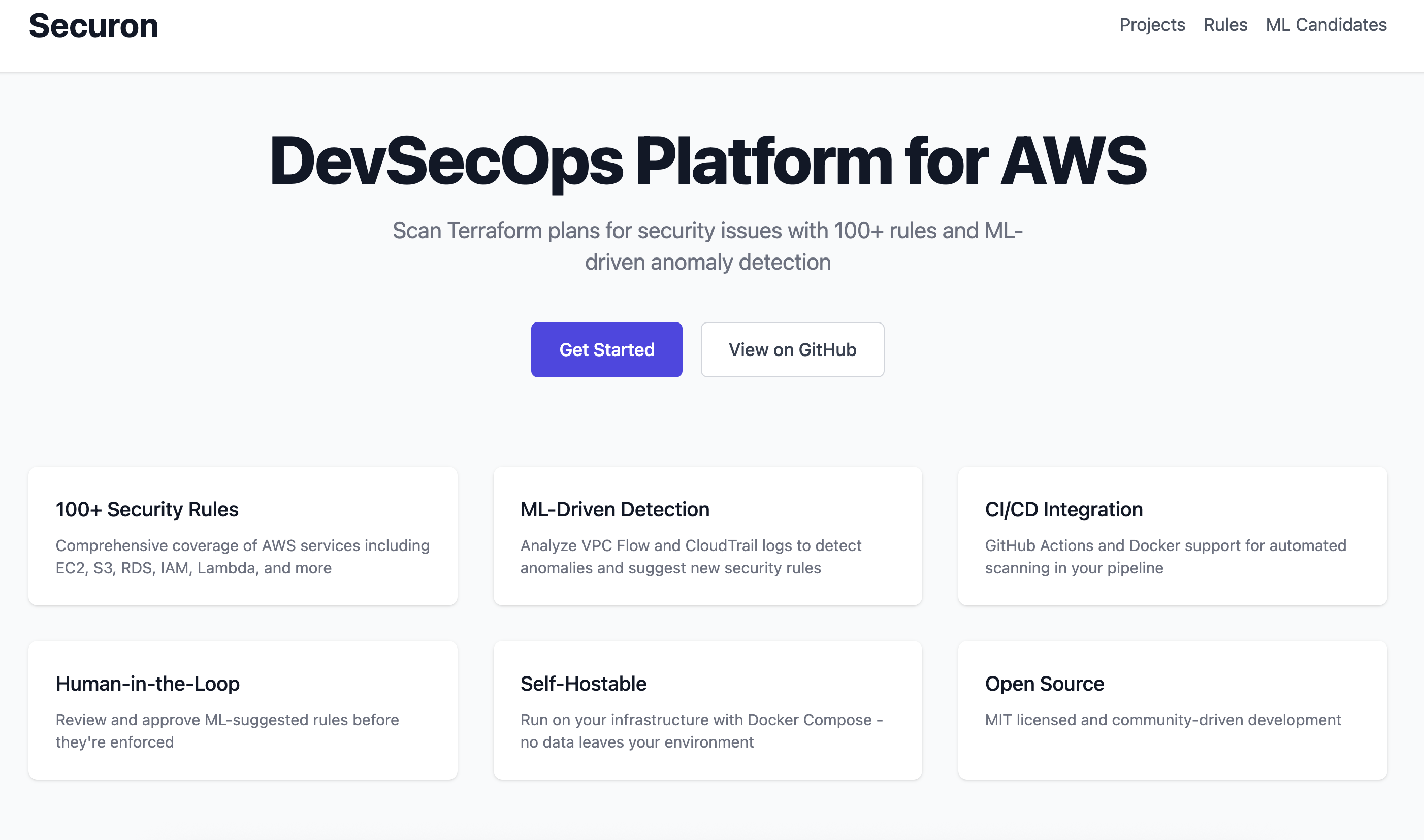


Fig. 3. Static Scan Results showing misconfigured Terraform resources detected before deployment.

This confirms the accuracy and practicality of the static analysis engine.

*B. Visualization of Active Security Rules*

The dashboard displayed all current IaC policies. This helped developers and security auditors understand what rules are enforced during each scan.

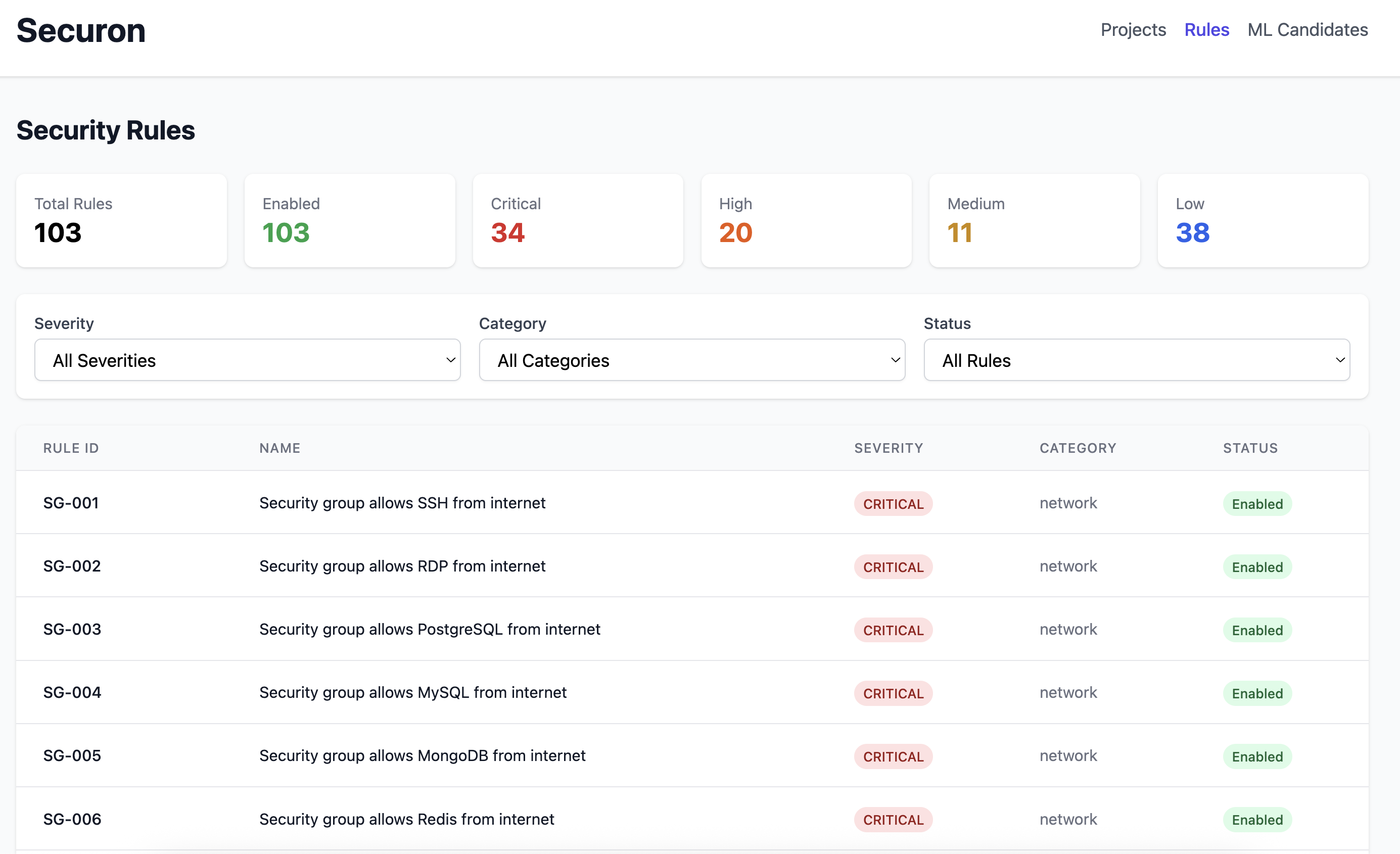


Fig. 4. Security Rules Dashboard listing active rules, severities, and categories.

This ensures transparency and auditability of the rule enforcement layer.

*C. Runtime AWS Log Monitoring*

Securon ingested live VPC Flow Logs and CloudTrail logs and displayed them in real time for analyst visibility. Malicious activities such as port scanning and repeated failed logins were clear during tests.

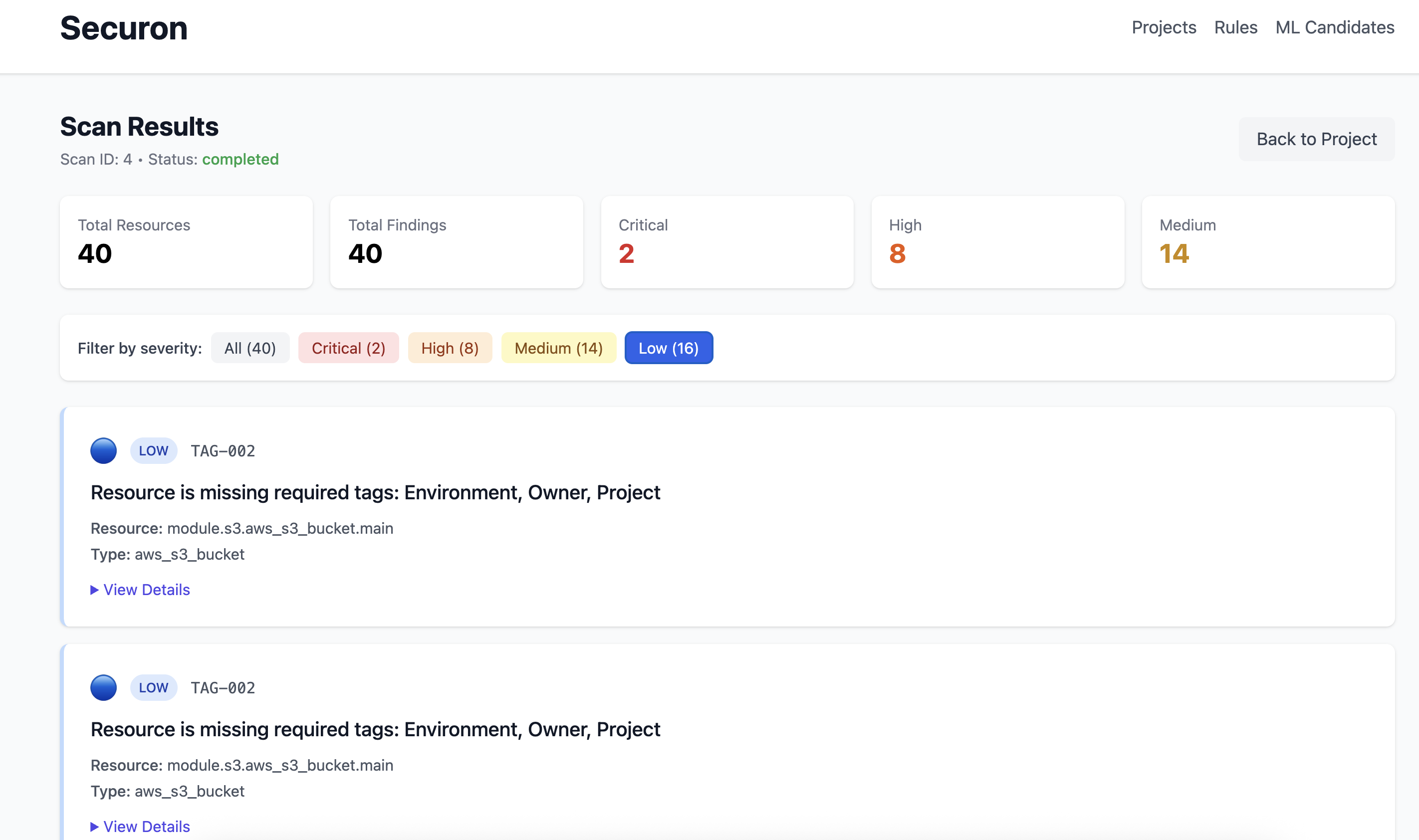


Fig. 5. Live Monitoring Interface showing real-time AWS VPC Flow Logs and CloudTrail events.

This verifies that Securon collects and streams AWS logs effectively.

*D. Machine Learning-Based Anomaly Detection*

Using preprocessed features, the Isolation Forest model detected anomalies such as unusual port activity, off-region IP access, and unusually high-volume traffic.

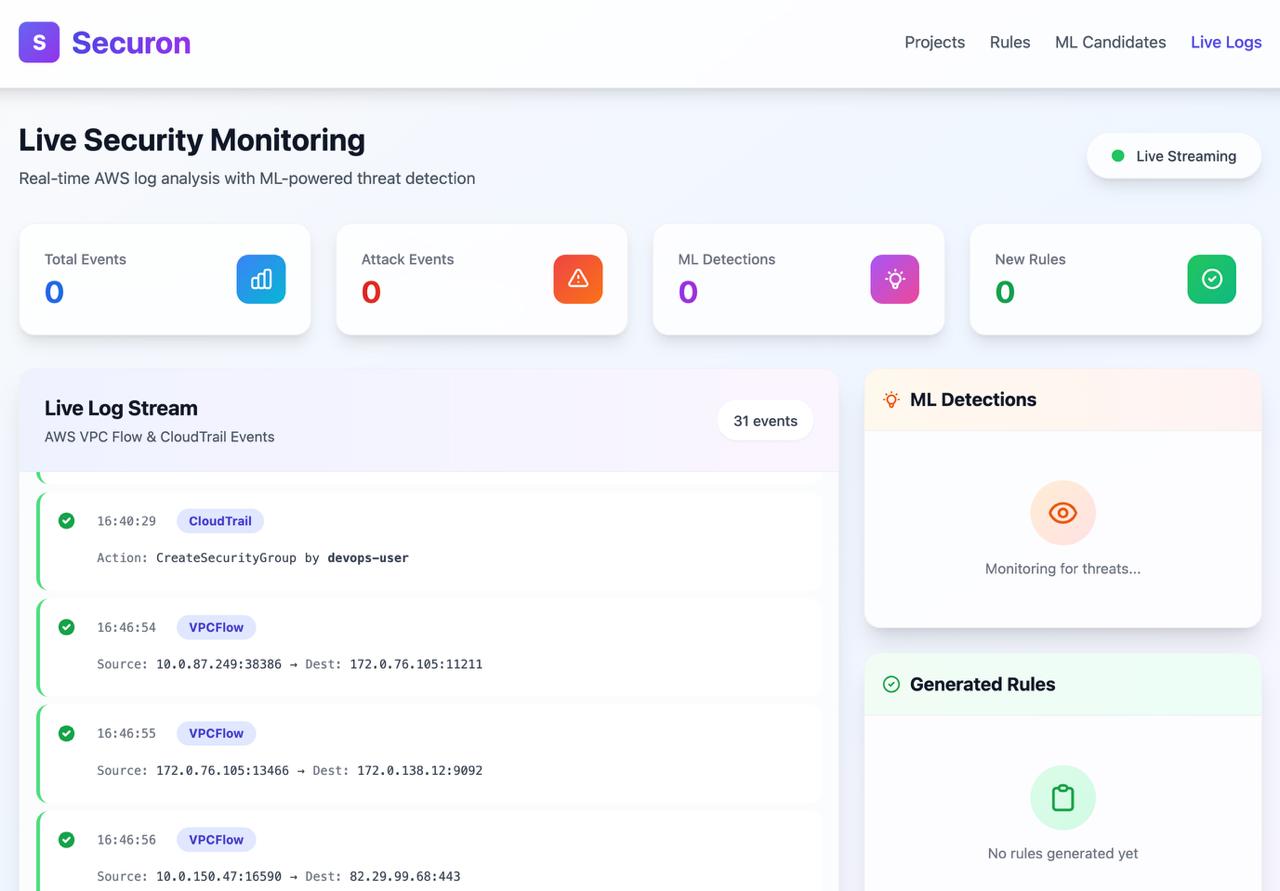


Fig. 6. ML Anomaly Detection Output highlighting outlier events detected by the Isolation Forest model.

These anomalies form the basis for generating adaptive rules.

*E. Terraform Plan Upload Interface*

The platform provides a simple way to upload execution plans manually, allowing pre-deployment scanning even outside CI/CD.

Fig. 7. Terraform Plan Upload Page used to initiate static scans on plan.json.

This supports ad-hoc validation workflows.

*F. ML Candidate Rule Generation*

The ML engine generated rule suggestions based on detected anomalies. These included recommendations to restrict certain ports and tighten IAM behavior.

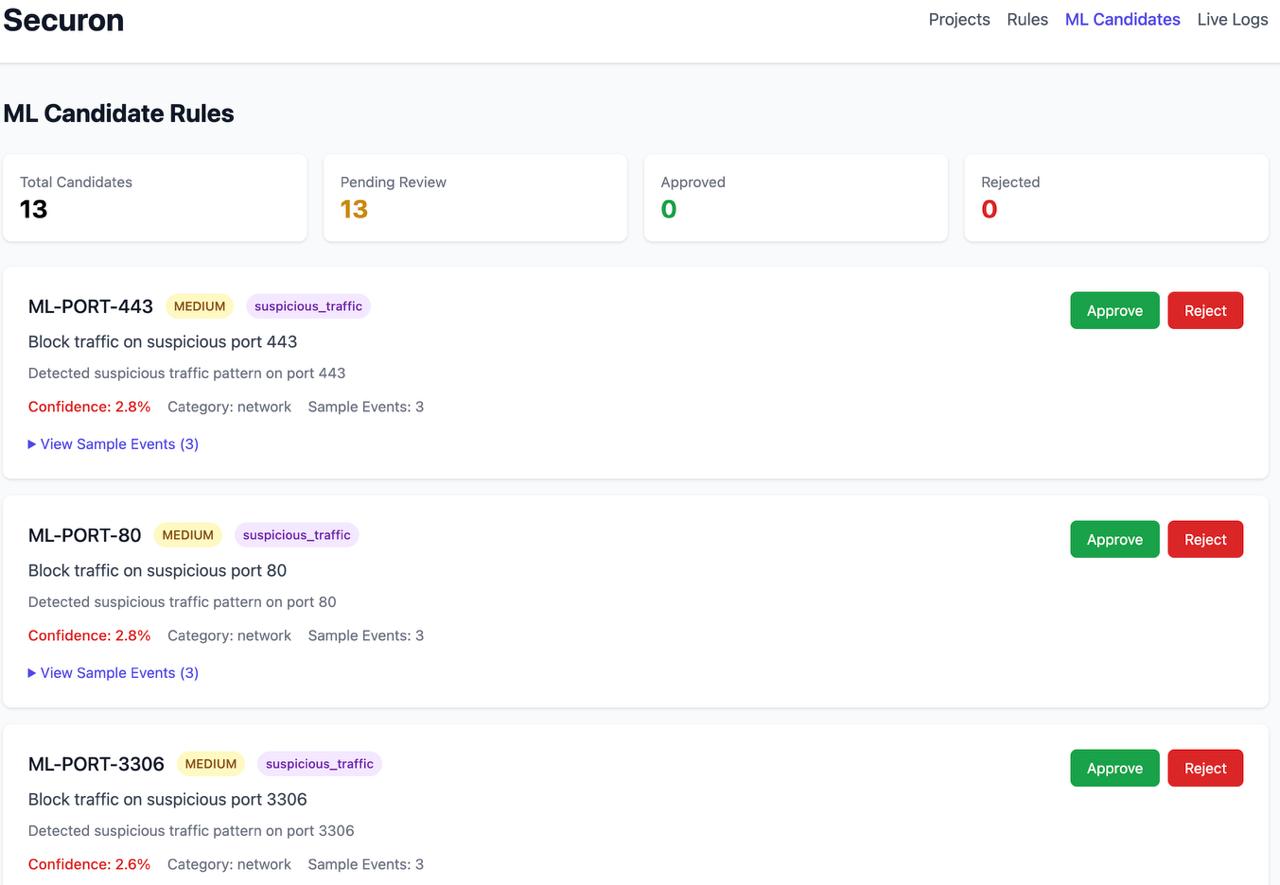


Fig. 8. ML Candidate Rules Screen showing automatically generated rule suggestions with confidence scores.

This confirms that the closed-loop adaptive model functions successfully.

*G. Discussion*

Results show that Securon effectively combines preventive and detective capabilities. The static scanner detects misconfigurations early, while the ML engine identifies runtime threats that static rules cannot predict. The rule-reinforcement mechanism ensures that previously detected attack behaviors cannot reoccur in future deployments. Although synthetic logs were used, real-world cloud telemetry will further improve detection quality and reduce false positives.

VI. LIMITATIONS AND FUTURE WORK

*A. Limitations*

The current evaluation relies on synthetic and semi-realistic AWS logs, which may not reflect all real-world challenges. Additionally, the framework focuses on AWS and Terraform-based environments.

*B. Future Enhancements*

Future work includes extending Securon to multi-cloud platforms, integrating advanced behavioral models like graph-based learning, enabling automated remediation for critical threats, and improving ML explainability for compliance purposes.

VII. CONCLUSION

This paper introduced Securon, a cloud security framework that combines pre-deployment IaC misconfiguration detection with post-deployment machine-learning-based anomaly analysis. By integrating Terraform plan scanning, AWS log monitoring, and adaptive rule generation into one workflow, Securon addresses the gap between preventive and detective cloud security methods.

The experimental evaluation showed that the static scanner effectively blocked insecure Terraform configurations before deployment. The ML engine detected runtime anomalies like port scans, unauthorized API actions, and abnormal access patterns. The human-in-the-loop approval process validated ML-derived rules before enforcement, making the reinforcement loop safe and effective.

While testing used synthetic AWS logs, real-world deployments with richer datasets should further improve anomaly detection accuracy and rule relevance. Future work aims to extend Securon to multi-cloud platforms, incorporate deeper behavioral models, and enable automated remediation for high-severity findings.Overall, Securon offers a practical approach to strengthen cloud security in DevSecOps environments, allowing for continuous learning and prevention of recurring vulnerabilities.

REFERENCES

[1] Amazon Web Services, “ AWS Security Best Practices, ” AWS Whitepaper, 2023.

[2] Amazon Web Services, “ VPC Flow Logs, ” AWS Documentation.( Online). Available https//docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html

[3] Amazon Web Services, “ AWS CloudTrail stoner companion, ” 2023.( Online). Available https// docs.aws.amazon.com/cloudtrail

[4] HashiCorp, “ Terraform Attestation, ” 2023.( Online). Available https// [www.terraform.io/docs](https://www.terraform.io/docs)

[5] Bridgecrew, “ Checkov structure as Code Static Analysis Tool, ” GitHub Repository.

[6] Aqua Security, “ tfsec Terraform Static Analysis, ” GitHub Repository.

[7] Accurics, “ Terrascan Detect Compliance and Security Violations Across IaC, ” GitHub Repository.

[8] Trend Micro, “ Cloud Misconfiguration Causes 65 of Cloud Breaches, ” Cloud Security Report, 2022.

[9] Palo Alto Networks, “ 2023 Unit 42 pall trouble Report. ”

[10] S. Alruheili and L. Xue, “ Cloud Security Risk from Misconfiguration A Methodical Study, ” IEEE Access, vol. 9, pp. 112934 – 112949, 2021.

[11] J. D. Ullman, “ Anomaly Discovery for Cloud Systems Using Unsupervised literacy, ” IEEE Deals on Cloud Computing, 2022.

[12] F. T. Liu et al., “ insulation timber, ” IEEE ICDM, pp. 413 – 422, 2008.

[13] M. Goldstein and S. Uchida, “ A relative Evaluation of Unsupervised Anomaly Discovery Algorithms for Multivariate Data, ” PLOS ONE, vol. 11, no. 4, 2016.

[14] A. G. Baydin, M. Pearce, et al., “ inimical Machine Learning for Cybersecurity, ” IEEE Security &amp; sequestration, 2021.

[15] S. Krishnan and Y. Chen, “ Log- grounded Anomaly Detection in Cloud structure Using Machine literacy, ” IEEE BigData, 2020.

[16] Elastic, “ Elastic Stack( ELK) Documentation, ” Elastic.co.

[17] Splunk Inc., “ Detecting pitfalls Using Splunk Enterprise Security, ” Splunk Docs, 2022.

[18] intellect exploration, “ 2023 State of Cloud Security Report. ”

[19] Lacework Labs, “ geste- Grounded pall trouble Discovery A Technical Analysis, ” 2022.

[20] S. M. Erfani et al., “ High- dimensional and large- scale anomaly discovery using a direct one- class SVM with deep literacy, ” Pattern Recognition, vol. 58, pp. 121 – 134, 2016.

[21] Google Cloud, “ Cloud Logging and Monitoring Best Practices, ” 2023.