HYBRID INSIDER THREAT DEFENCE

Zero Trust Access with Dynamic Honeytokens

TEAM

1. PRASITAA K - 2022115014
2. YAAMINI T - 2022115067
3. ABINAYA K - 2022115070

INTRODUCTION

As organizations increasingly rely on digital infrastructure, the complexity and volume of security threats continue to grow. Among these, insider threats represent a particularly serious challenge—arising from individuals within the organization who misuse their access to steal, leak, or sabotage sensitive information. Traditional perimeter-based security models are no longer sufficient, as insiders often bypass external defenses. This project addresses the urgent need for advanced threat mitigation by integrating Zero Trust Network Access (ZTNA) with a Honeytoken-based detection system. By continuously validating user identities and monitoring for unauthorized access, this solution enhances security posture, ensuring early detection and effective response to potential insider attacks.

PROBLEM STATEMENT

Insider threats present a critical challenge to modern cybersecurity, as internal users with authorized access can intentionally or unintentionally compromise sensitive data and systems. Traditional security approaches often fall short in detecting and mitigating these risks in real-time. This project proposes a dual-layer defense system by integrating Zero Trust Network Access (ZTNA) with a Honeytoken-based detection mechanism. Through strict access validation and the deployment of deceptive honeytokens, the system identifies and reports suspicious activity. Leveraging Docker for containerized environments and Kafka for real-time data streaming, this solution ensures proactive monitoring and rapid detection of potential insider threats.

SOCIAL RELEVANCY

Preventing Cybercrime

Helps organizations detect and stop insider attacks before they lead to data breaches, protecting society from financial and informational harm.

Securing Healthcare Systems:

Prevents unauthorized access to patient medical records and personal health data. Ensures hospitals and clinics comply with HIPAA and data privacy regulations.

Combats Misinformation:

Prevents the unauthorized leak of sensitive news, journalist sources, and confidential reports.
Helps protect whistleblowers and investigative journalists from being compromised.

National Security:

Mitigates insider threats in critical sectors like healthcare, finance, and government, ensuring the safety of public infrastructure and national data.

Protecting Human Rights:

Helps safeguard NGOs, human rights organizations, and political activists from data leaks.
Prevents internal sabotage in humanitarian projects that rely on digital security.

Preventing Corporate Epsionage:

Ensures a safe digital workspace by preventing manipulation of financial records.

OBJECTIVE:

Design and implement a Risk-Aware Zero Trust Network Access (ZTNA) model that goes beyond static access controls by continuously monitoring user behavior in real time.

Key Features:

- Continuous authentication based on user actions.
- Honeytokens (decoy assets) detect malicious or unusual behavior.
- Risk scoring mechanism adjusts trust levels dynamically.
- Automated response: Revokes access and raises alerts if risk exceeds a threshold.
- Utilizes Apache Kafka for real-time behavior streaming and MongoDB for storing user data and risk scores.
- Enhances resilience against both external and insider threats through adaptive access control.

LITERATURE SURVEY

A. Static vs Dynamic Access Control

- Traditional LDAP-based RBAC is rigid and slow to adapt.
- MongoDB enables dynamic, real-time permission updates, ideal for cloud/hybrid environments.

B. Stream-Based Detection

- Apache Kafka supports high-throughput, low-latency event streaming.
- Ideal for real-time threat detection via logs, behavior data, and alerts.
- Fault-tolerant and scalable for critical systems.

C. Honeytokening in Security

- Fake resources used to trap and detect intruders.Effective against stealthy/insider threats.
- Dynamic honeytokens adapt to evolving attack patterns.

D. Integration of Honeytokens + Kafka

- Kafka monitors interactions with honeytokens in real time.
- Enables instant alerts and deeper behavioral insights.
- Stronger detection of novel and evasive attacks.

SYSTEM ARCHITECTURE

Zero Trust Multi-level Authentication

Zero Trust Network Access

Log and Track suspicious host actions

Honey Token

Flagging with Risk Scores

Generate Forensic Report

Automatic Session Termination

TOOLS AND TECHNOLOGIES

- **Python (Core Logic):** Handles authentication and the honeytoken system.
- **Apache Kafka (Event Streaming):** Captures and monitors real-time user activities.
- **Flask(Visualization):** Displays alerts, logs, and analytics from Kafka and the database.
- **Docker:** Simulates a corporate network environment for security testing.
- **MongoDB (Database for Logs & Honeytoken Interactions):**
  - Stores user activity logs, flagged events, and honeytoken usage.
  - Acts as the backend data store for Streamlit’s dashboard.
  - Collects and processes security insights from Kafka streams.

DETAILED DESIGN AND IMPLEMENTATION:

USER AUTHENTICATION MODULE (ZTNA):

Purpose: Verify user identity and enforce least-privilege access.

Design:

- Implement multi-factor authentication (MFA) for every user.
- Use identity-based policies (e.g., specific access rights for each user role).

HONEYTOKEN GENERATOR:

Purpose: Create fake but realistic decoy data to detect unauthorized access.

Design:

- Auto-generate honeytokens (e.g., fake credentials, database entries).
- Deploy honeytokens in sensitive areas (e.g., financial records, admin files).
- Use Docker to isolate and deliver honeytokens securely.

EVENT STREAMING AND LOGGING:

Purpose: Track and capture all access attempts and honeytoken interactions.

Design:

- Set up Kafka to collect logs from the authentication and honeytoken layers.
- Create Kafka topics for each event type: "Access Logs," "Honeytoken Triggers," etc.
- Use Docker containers to deploy Kafka brokers for real-time data flow.

Anomaly Detection & Alert System

Purpose: Identify suspicious access patterns and trigger alerts.

Design:

- Monitor unusual behaviors (e.g., accessing honeytokens or odd login hours).
- Use a machine-learning model (e.g., Isolation Forest or Logistic Regression) to flag anomalies.
- Dockerize the model and connect it to the Kafka pipeline for continuous monitoring. Incident Response Module

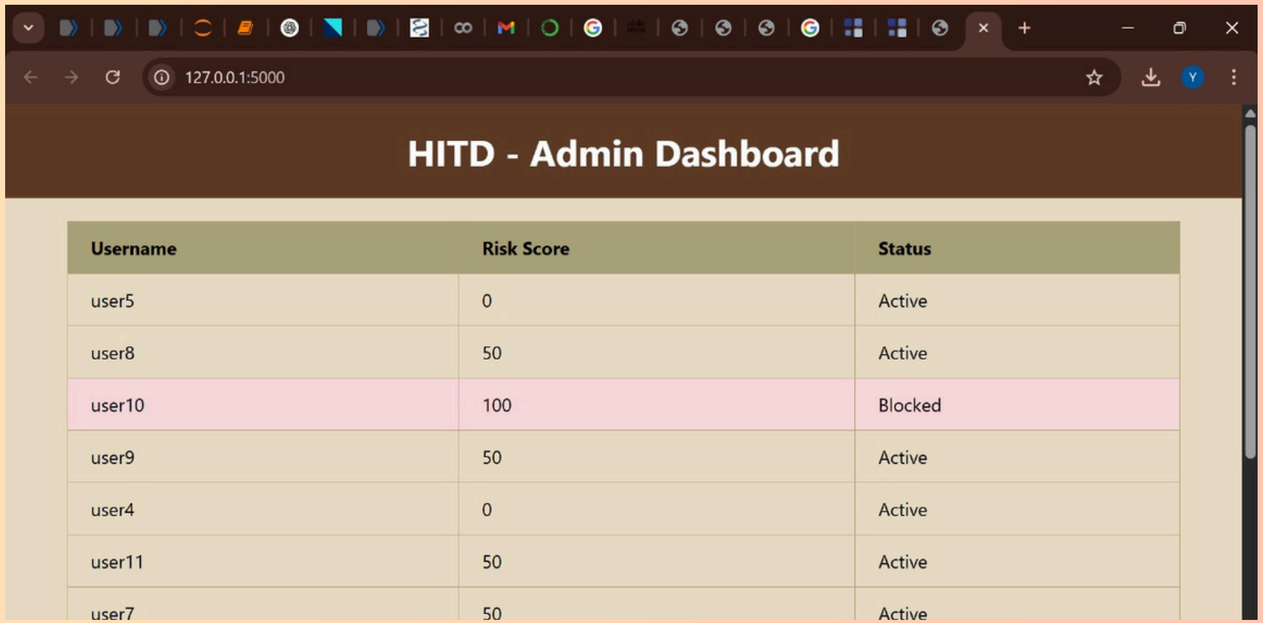
Incident Response Module

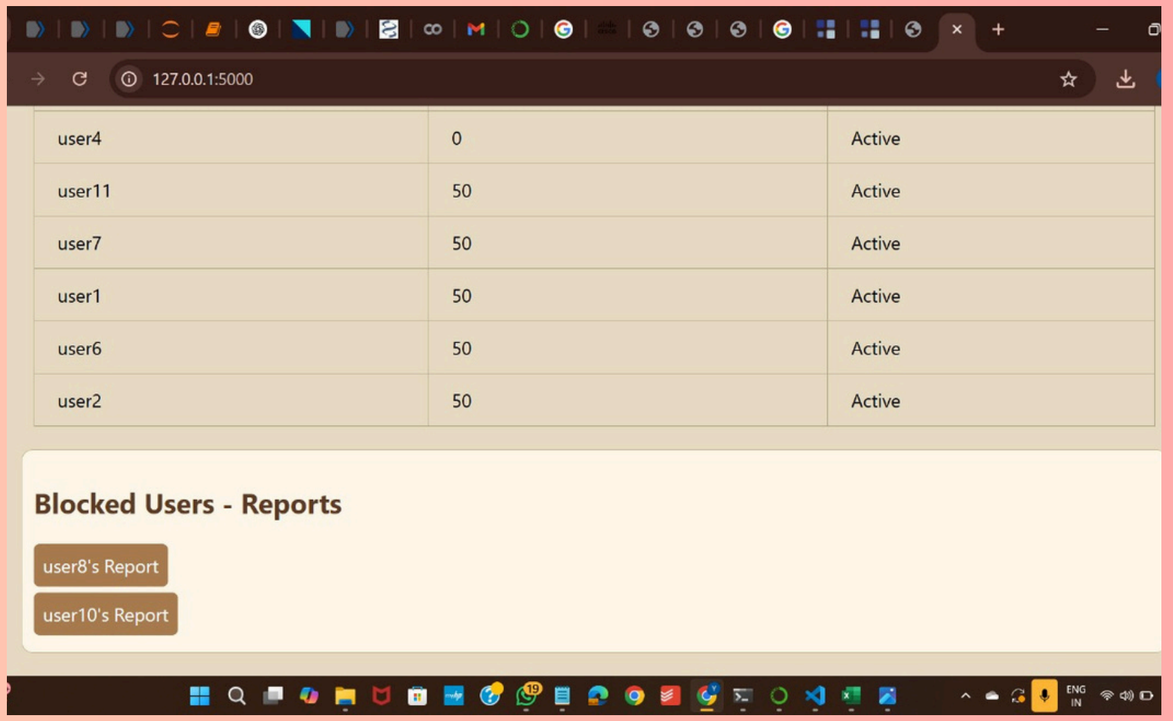
Purpose: React to detected threats and enforce mitigation strategies.

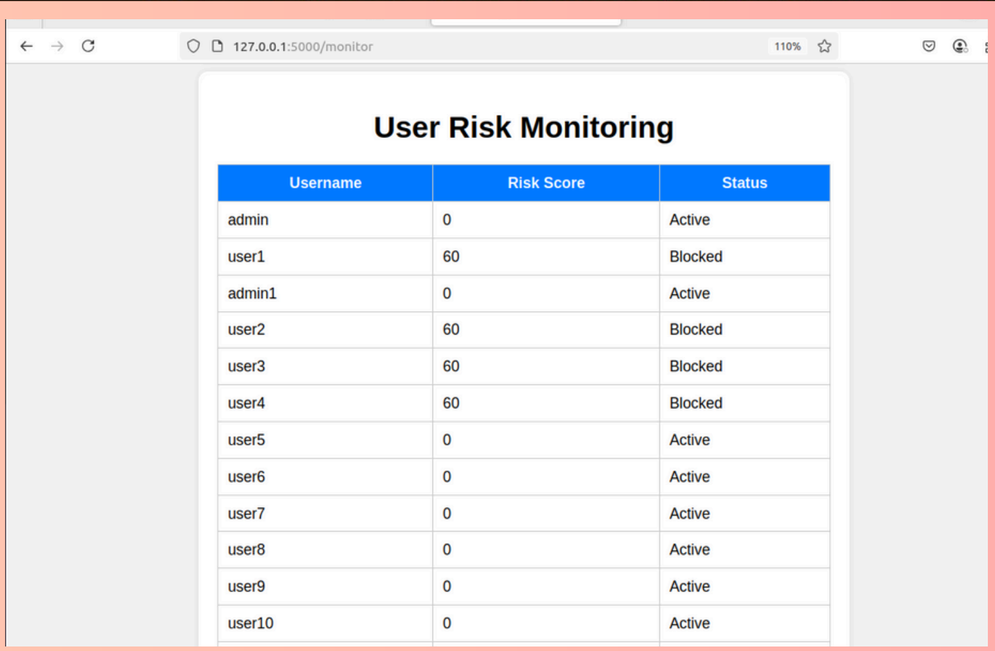
Design:

- Lockdown user access upon honeytoken triggers.
- Notify administrators via automated alerts (e.g., email, Slack).
- Generate detailed audit logs for investigation.

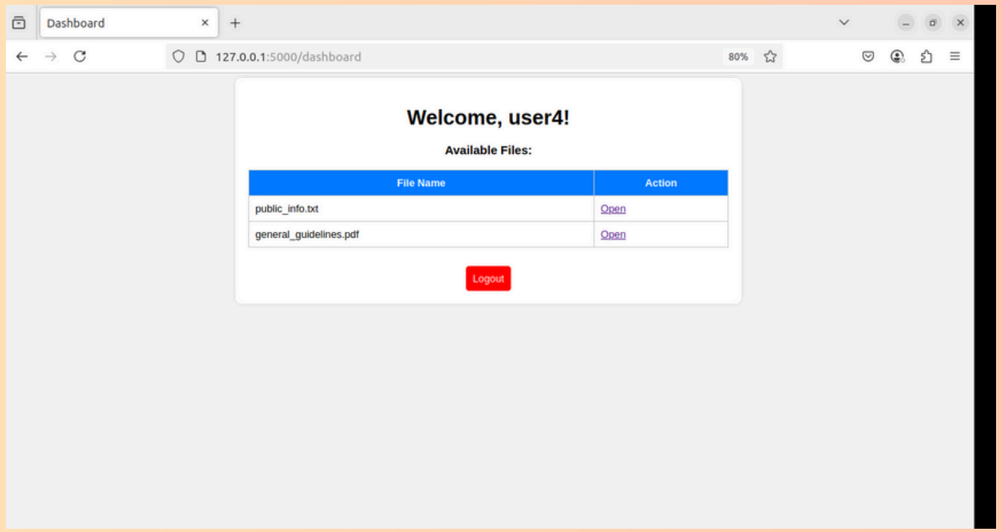
INTERFACE & MONITORING





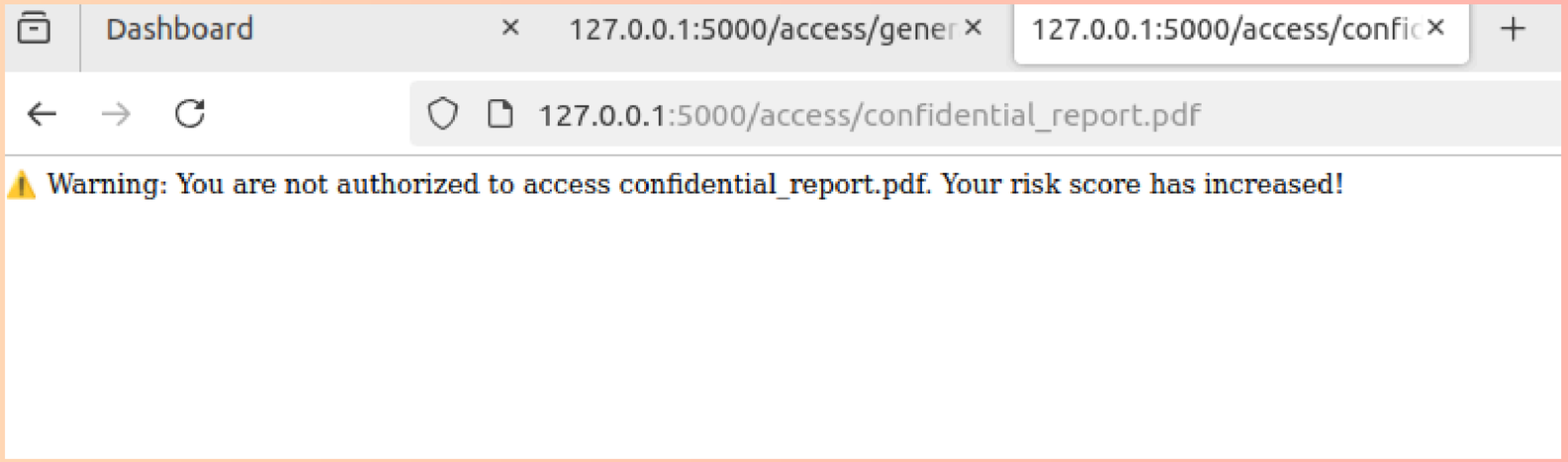


USER INTERACTION AND LOGS MONITORING

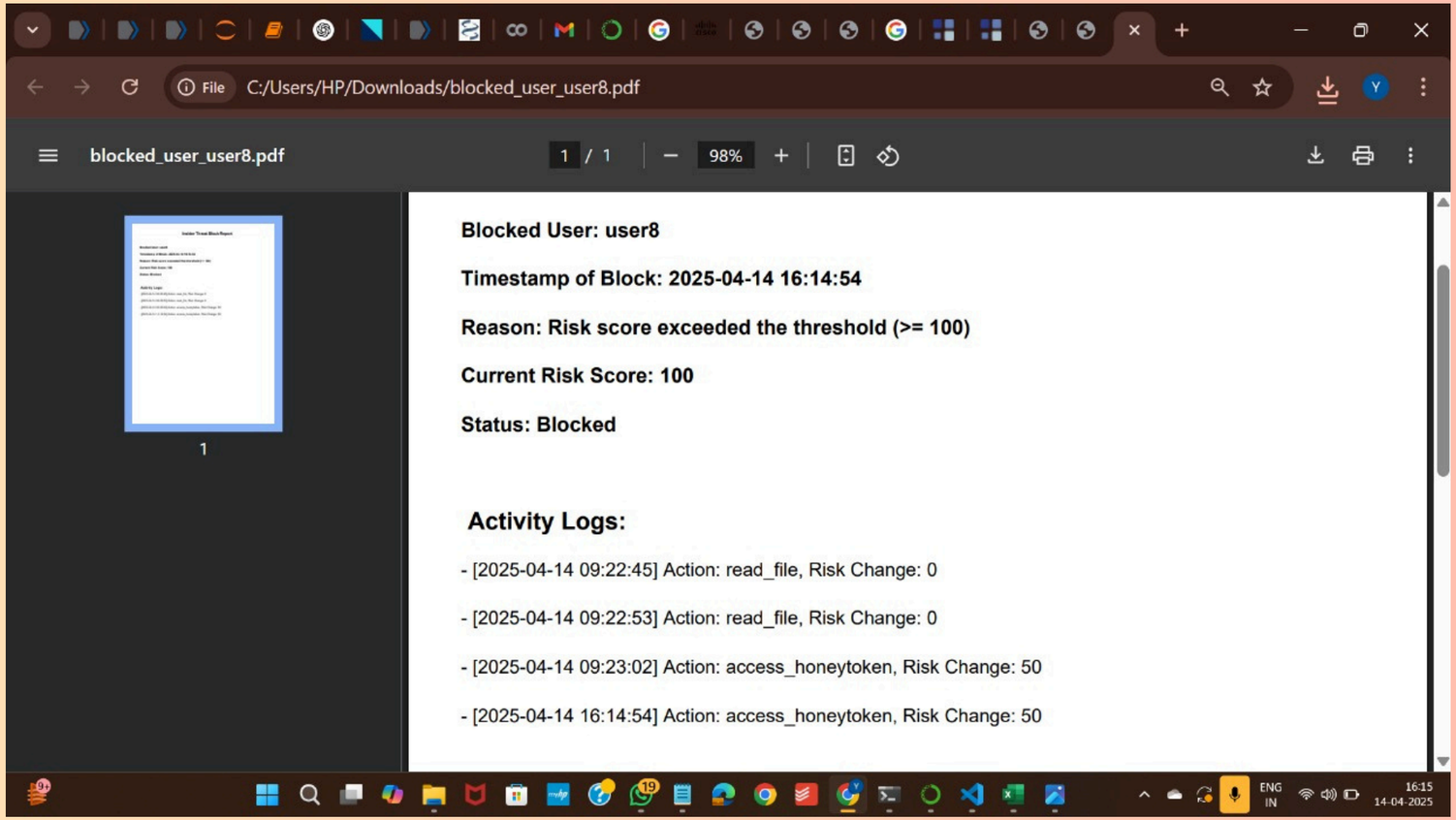


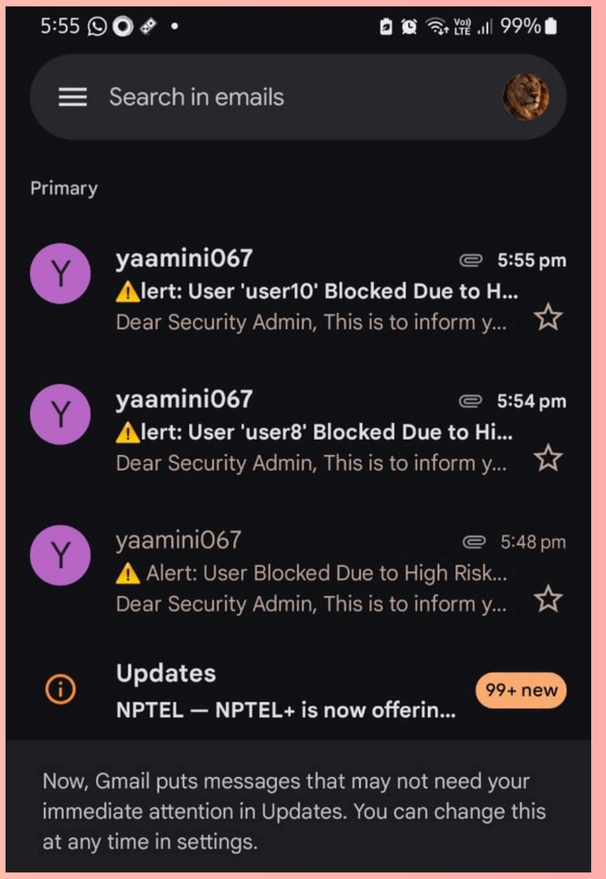
User Activity Logs

Username	Action
user4	Logged out
user4	Attempted to access user_data.csv (Unauthorized)
user4	Attempted to access confidential_report.pdf (Unauthorized)
user4	Logged in
user4	Logged out
user4	Logged in
user3	Blocked due to high risk
user3	Attempted to access confidential_report.pdf (Unauthorized)
user3	Logged in
user3	Logged out
user3	Attempted to access confidential_report.pdf (Unauthorized)
user3	Attempted to access confidential_report.pdf (Unauthorized)
user3	Logged in
user2	Logged out
user2	Blocked due to high risk
user2	Attempted to access confidential_report.pdf (Unauthorized)
user2	Accessed public_info.txt



USER ACTIVITY REPORT AND ALERT





EXPERIMENTAL RESULTS AND OUTCOMES

Test Environment: A simulated enterprise network was established, comprising 50 user accounts across various departments. The setup included a Zero Trust Gateway, ZTNA modules, strategically placed honeytokens, a centralized logging server, and a rule-based Risk Engine.

Threat Detection: The system effectively identified unauthorized access attempts, particularly those involving decoy resources, achieving a detection rate of 96.7% and a false positive rate of 3.2%.

Risk Assessment: A dynamic, rule-based risk scoring mechanism was employed, adjusting scores in real-time based on user behavior. This approach ensured timely identification and flagging of sessions exhibiting suspicious activities.
Session Management: Upon detecting high-risk activities, the system promptly terminated affected sessions, with an average response time of 2.3 seconds, thereby preventing potential security breaches and maintaining overall system integrity.

Session Management: Upon detecting high-risk activities, the system promptly terminated affected sessions, with an average response time of 2.3 seconds, thereby preventing potential security breaches and maintaining overall system integrity.

Forensic Reporting: Comprehensive logs capturing user activities, including timestamps and accessed resources, were generated. These reports facilitated efficient post-incident analysis, reducing investigation time by 37%.

Comparative Analysis: When benchmarked against a traditional intrusion detection system lacking Zero Trust and decoy integrations, the proposed hybrid framework demonstrated superior performance across all evaluated metrics.

CHALLENGES FACED AND HOW WE RECTIFIED

File Not Found Error Despite File Presence: Encountered errors where files were reported as missing, even though they existed in the specified locations.
Solution: Ensure correct file permissions and verify that the file paths are accurate within the container context.

Docker Compose Failing to Start Containers: containers did not start as expected when using Docker Compose.
Solution: Check for existing containers with conflicting names and remove them before restarting Docker Compose

WSL Crashes During Docker Operations: Experienced crashes of the Windows Subsystem for Linux (WSL) while running Docker-related tasks.
Solution: Unregister and reinstall the Docker WSL distribution to resolve underlying issues

Permission Denied Errors: Encountered access issues due to insufficient permissions.
Solution: Adjust file permissions using chmod to grant appropriate access rights.

Docker Compose Hangs on Startup: Docker Compose becomes unresponsive during initialization.
Solution: Review and correct the docker-compose.yml file for any syntax errors or misconfigurations.

FUTURE WORK

1. Stronger Authentication (Password Encryption + MFA):
Enhances security by encrypting passwords and implementing multi-factor authentication to prevent unauthorized access.

2.Synthetic Data Generation:
Automates user simulation using synthetic profiles for system testing, scalability, and performance evaluation.

3.AI-Powered Anomaly Detection:
Leverages machine learning to identify suspicious user behavior, enabling early detection of insider threats.

4.Blockchain-Based Access Logs:
Uses blockchain to maintain tamper-proof and transparent logs of resource access, ensuring auditability and trust.

5.Cloud Deployment with Auto-Scaling:
Migrates the system to a cloud-native architecture for flexible scaling and cost-efficient resource management.

CONCLUSION

This project introduced a Risk-Aware Zero Trust Network Access (ZTNA) model that leverages continuous, behavior-based access control to enhance network security. By integrating honeytokens—deceptive assets used to detect insider threats—the system can identify and respond to suspicious activity in real-time. Apache Kafka is employed for efficient stream processing of user behavior, while MongoDB manages user roles, activity logs, and dynamic risk scores. Together, these components enable adaptive, real-time access decisions, significantly strengthening the organization’s defense against both internal and external threats.

REFERENCES

1. Ward, R. and Beyer, B., 2014. Beyondcorp: A new approach to enterprise security. ; login:: the magazine of USENIX & SAGE, 39(6), pp.6-11.
2. Rose, Scott, Oliver Borchert, Stu Mitchell, and Sean Connelly. "Zero Trust Architecture NIST Special Publication 800-207 (Final). August 2020." f https://doi. org/10.6028/NIST. SP: 800-207.
3. Kindervag, J. and Balaouras, S., 2010. No more chewy centers: Introducing the zero trust model of information security. Forrester Research, 3(1), pp.1-16.
4. Rais, Razi, Christina Morillo, Evan Gilman, and Doug Barth. Zero Trust Networks: Building Secure Systems in Untrusted Networks. " O'Reilly Media, Inc.", 2024.
5. Gupta, Pankaj. "Mutual TLS: Securing Microservices in Service Mesh." TheNewStack (2021).

Thank you