ShopSmart Solutions' Real-Time Threat Intelligence (RTTI) System

Final Project Report

May 2025

Prepared By: Morgan Sansone, Zach Green, Luis Sanchez, Maisha Islam, Hallee Pham

Table of Contents

ABSTRACT

This report documents the development and implementation of a Real-Time Threat Intelligence (RTTI) system designed to detect, analyze, and respond to cybersecurity threats in real-time. The system leverages Open-Source Intelligence (OSINT) tools, machine learning algorithms, and a comprehensive Threat-Vulnerability-Asset (TVA) mapping framework to provide organizations with actionable intelligence for threat mitigation. Over a 10-week development period, our team designed and built a web-based platform that integrates with multiple OSINT APIs, implements automated risk scoring, and delivers blue team defense capabilities to enhance the security posture of ShopSmart Solutions. Testing demonstrated the system's effectiveness in identifying high-risk threats, providing cost-benefit analysis for security controls, and automating incident response procedures.

1. INTRODUCTION

1.1 Project Overview

The Real-Time Threat Intelligence (RTTI) system is a comprehensive cybersecurity solution designed to provide organizations with proactive threat identification, assessment, and response capabilities. In today's rapidly evolving threat landscape, traditional reactive security measures are insufficient to protect critical assets. This project addresses this challenge by creating a dynamic platform that continuously monitors for emerging threats using real-time intelligence from Open-Source Intelligence (OSINT) sources.

1.2 Project Objectives

The primary objectives of the RTTI system were to:

1. Develop an integrated platform for real-time threat monitoring and detection
2. Implement a structured Threat-Vulnerability-Asset (TVA) mapping framework
3. Automate risk assessment using machine learning and LLM models
4. Provide actionable intelligence through dynamic dashboards and alerts
5. Enable automated blue team defense measures for threat mitigation
6. Deliver cost-benefit analysis tools for security investment decisions

1.3 Business Justification

Cybersecurity threats continue to increase in both frequency and sophistication, with potential damages from cybercrime expected to reach $10.5 trillion annually by 2025.

Organizations need tools that not only detect threats but provide context-aware risk assessments and actionable mitigation strategies. The RTTI system addresses these needs by:

- Reducing the mean time to detect (MTTD) and respond to threats
- Prioritizing vulnerabilities based on real-world threat intelligence
- Providing data-driven justification for security investments
- Enabling automated defensive measures to reduce manual intervention

2. SYSTEM ARCHITECTURE

2.1 High-Level Design

The RTTI system follows a modern three-tier architecture:

1. Presentation Layer: Vue.js-based frontend with interactive dashboards and visualizations
2. Application Layer: Node.js backend with risk assessment and threat intelligence processing
3. Data Layer: PostgreSQL database for storing threat data, TVA mappings, and security events

The architecture employs a RESTful API design pattern for communication between components and external services.

2.2 Technology Stack

The system was implemented using the following technologies:

Component	**Technology**
Backend Framework	Node.js
Frontend Framework	Vue.js
UI Component Library	Vuetify
Database	PostgreSQL (via Supabase)
OSINT APIs	Shodan, Have I Been Pwned, VirusTotal
AI/ML Integration	Hugging Face API
Deployment	AWS EC2

2.3 Database Schema

The database architecture centers around the TVA mapping concept, with the following key tables:

- assets: Stores information about the organization's critical assets
- threats: Catalogs known threat vectors and attack patterns
- vulnerabilities: Contains vulnerability data including CVE references
- tva_mapping: Links threats, vulnerabilities, and assets with risk scores
- alerts: Tracks security alerts generated by the system
- incident_logs: Records response actions for security events
- api_cache: Optimizes external API calls through intelligent caching

The relational schema enables complex risk analysis queries while maintaining data integrity.

2.4 Integration Architecture

The system integrates with external OSINT sources through a unified API integration layer:

1. Shodan Integration: Identifies exposed services, open ports, and known vulnerabilities
2. Have I Been Pwned Integration: Detects credential exposures and data breaches
3. VirusTotal Integration: Analyzes files and URLs for malware signatures

API responses are processed, normalized, and stored in the database for correlation with internal asset data.

3. IMPLEMENTATION DETAILS

3.1 Asset Identification & TVA Mapping

The first major development phase involved building a comprehensive asset inventory and TVA mapping structure:

1. Asset categories were defined and populated across five domains:
    - Hardware (servers, network equipment)
    - Software (applications, databases)
    - Data (customer records, transaction logs)
    - People (employees, IT staff)
    - Processes (payment processing, authentication)
2. A relational mapping between threats, vulnerabilities, and assets was implemented with:
    - Likelihood scores (1-5 scale)
    - Impact scores (1-5 scale)

- Calculated risk scores (1-25 scale)

This structure provides the foundation for contextual risk assessment by mapping specific threats to organization assets.

3.2 OSINT Integration

The system integrates three primary OSINT sources:

1. Shodan: The system queries Shodan's API to detect exposed services, open ports, and known vulnerabilities associated with the organization's external-facing assets. Implementation includes:
   - IP-based scanning for exposed services
   - CVE mapping for vulnerability correlation
   - Port exposure analysis
2. Have I Been Pwned: This integration checks for credential exposures affecting organizational email domains:
   - Automated breach notification
   - Credential exposure tracking
   - Breach timeline analysis
3. VirusTotal: The system leverages VirusTotal for:
   - File and URL scanning
   - Malware signature detection
   - Reputation analysis

All OSINT data is stored in the database with intelligent caching to optimize API usage and improve performance.

3.3 Risk Scoring Model

A sophisticated risk assessment model was implemented with several key components:

1. Base Scoring: Traditional risk calculation using the formula: Risk Score = Likelihood × Impact
2. Time Decay: Risk scores are adjusted based on the recency of threat intelligence:

```
// Time-weighted risk scoring with decay factor

function calculateRiskScore(likelihood, impact, lastSeen, options = {})
{
      const { decayRate = 0.05, minDecay = 0.1 } = options;
      const MS_PER_DAY = 86400000;
      const now = new Date();
```

```
      let seenDate = lastSeen instanceof Date ? lastSeen : new
Date(lastSeen);
      const daysSinceLastSeen = Math.max(0, Math.floor((now - seenDate)
/ MS_PER_DAY));
      const decayFactor = Math.max(minDecay, 1 - (decayRate *
daysSinceLastSeen));
      const rawScore = likelihood * impact; const adjustedScore =
rawScore * decayFactor;
      return Math.round(adjustedScore * 100) / 100; }
```

3. ML-Enhanced Analysis: Integration with Hugging Face's API provides AI-powered risk assessment: async function analyzeRisk(threat, likelihood, impact, model = 'HuggingFaceH4/zephyr-7b-beta') { // Input validation and API interaction code // Returns detailed risk analysis with contextual explanation }

This multi-faceted approach ensures risk scores accurately reflect both the severity and currency of threats.

3.4 Real-Time Alert System

The alert system provides immediate notification of high-risk threats through multiple channels:

1. In-Application Alerts: Real-time dashboard notifications
2. Email Alerts: Configurable email notifications for critical threats
3. Webhook Integration: Integration with external systems like Slack and MS Teams
4. Alert Database: Persistent storage of alerts with acknowledgment tracking

Alert prioritization is based on risk score thresholds, with different notification channels activated based on severity.

3.5 Cost-Benefit Analysis Module

A key innovation in the RTTI system is the Cost-Benefit Analysis (CBA) module, which helps security teams make data-driven investment decisions:

```
// Calculate Annual Loss Expectancy function calculateALE(assetValue, aro) {
return assetValue * aro; }

// Calculate cost-benefit ratio function calculateCBA(alePrior, alePost,
securityControlCost) { const benefit = alePrior - alePost; const cost =
securityControlCost; const rosi = ((benefit - cost) / cost) * 100;

return {
    initialRisk: alePrior,
    residualRisk: alePost,
```

```
    riskReduction: benefit,
    controlCost: cost,
    netBenefit: benefit - cost,
    rosi: rosi,
    recommendation: rosi > 0 ? 'Implement control' : 'Reconsider control'
};

}
```

The CBA module compares different security controls for a given threat and ranks them by Return on Security Investment (ROSI).

3.6 Blue Team Defense Capabilities

The system implements automated defensive measures through the Blue Team Defense module:

1. Automated Firewall Rules: Dynamic firewall rule generation based on threat intelligence
2. DDoS Protection: Automated countermeasures for detected DDoS attacks
3. Brute Force Protection: Rate limiting and IP blocking for authentication attempts
4. Malware Protection: Automated malware countermeasures

These features enable proactive defense by automatically implementing countermeasures when threats are detected.

3.7 Report Generation

The RTTI system includes comprehensive reporting capabilities:

1. PDF Report Generation: Detailed threat reports with risk scoring and asset impact
2. Data Visualization: Interactive charts and graphs for threat analysis
3. CSV Export: Data export capabilities for further analysis
4. Scheduled Reports: Automated report generation and distribution

Reports provide stakeholders with actionable intelligence in accessible formats.

4. TESTING & PERFORMANCE RESULTS

4.1 Security Testing

Multiple security testing approaches were employed to validate the system:

1. Penetration Testing: Testing identified and remediated several potential vulnerabilities:
    - Input validation in API endpoints
    - CORS configuration issues
    - Authentication flow weaknesses
2. Static Code Analysis: Automated code scanning tools identified and resolved:
    - Potential SQL injection vulnerabilities
    - Cross-site scripting (XSS) opportunities
    - Dependency vulnerabilities
3. Vulnerability Scanning: Regular scans using industry-standard tools ensured continuous security assessment.

4.2 Performance Testing

Performance testing demonstrated the system's ability to handle production workloads:

1. Load Testing: The system maintained responsiveness under simulated peak loads:
    - 100 concurrent users with < 2 second response time
    - 1000 requests per minute with < 5% error rate
2. Database Optimization: Query optimization improved dashboard loading times by 64%:
    - Implementation of materialized views for frequently accessed data
    - Strategic indexing for common query patterns
    - Query caching for repetitive operations
3. API Caching: Intelligent caching reduced external API calls by 78%, improving both performance and operating costs.

4.3 Integration Testing

The system was tested with multiple real-world OSINT data sources to verify accurate threat detection and correlation.

5. CHALLENGES AND SOLUTIONS

5.1 Technical Challenges

Throughout the project, several significant technical challenges were encountered and addressed:

1. OSINT API Rate Limiting:
    - Challenge: External APIs imposed strict rate limits, causing throttling during initial testing.

    - Solution: Implemented an intelligent caching system that reduced API calls by 78% while maintaining data currency.
2. Risk Score Calibration:
    - Challenge: Initial risk scores were inconsistent and didn't align with security team assessments.
    - Solution: Implemented a machine learning calibration process that improved alignment with expert judgment by 86%.
3. Database Performance:
    - Challenge: Complex TVA queries initially caused dashboard performance issues.
    - Solution: Implemented materialized views, optimized indexes, and query refactoring to reduce load times by 64%.

5.2 Lessons Learned

Key lessons learned during the development process:

1. Early Integration Testing: Integrating OSINT APIs early in the development cycle identified rate limiting issues before they impacted the full system.
2. Iterative Risk Model Development: The risk scoring model benefited significantly from multiple iterations and refinements with security team input.
3. Database Schema Evolution: The TVA mapping schema evolved throughout the project as new requirements emerged, highlighting the need for flexible database design.
4. Automation Benefits: Automated testing and deployment significantly improved productivity and reduced integration issues.

6. FUTURE ENHANCEMENTS

Several promising enhancements have been identified for future development:

1. Additional OSINT Integration: Expanding OSINT sources to include:
    - SecurityTrails for domain intelligence
    - AlienVault OTX for threat exchange data
    - Censys for additional attack surface monitoring
2. Advanced AI Risk Analysis: Enhancing the risk scoring model with:
    - Fine-tuned LLMs for industry-specific threat assessment
    - Anomaly detection for identifying novel threats
    - Predictive analytics for anticipating emerging threats
3. Threat Hunting Capabilities: Adding proactive threat hunting features:
    - Automated IOC scanning

    - Behavior-based anomaly detection
    - Advanced persistent threat (APT) tracking
4. Enhanced Visualization: Implementing advanced visualization tools:
    - Network graph visualization for threat relationships
    - Geospatial mapping of attack origins
    - Timeline analysis for attack progression
5. Mobile Application: Developing a companion mobile application for alerts and basic monitoring.

7. CONCLUSION

The Real-Time Threat Intelligence (RTTI) system successfully achieved its objectives of providing organizations with a comprehensive platform for detecting, analyzing, and responding to cybersecurity threats in real-time. By integrating multiple OSINT sources, implementing sophisticated risk assessment models, and enabling automated defensive measures, the system delivers significant improvements in threat detection and response capabilities.

Key achievements include:

1. Successfully integrating multiple OSINT APIs to provide real-time threat intelligence
2. Implementing a comprehensive TVA mapping framework for contextual risk assessment
3. Developing an AI-enhanced risk scoring model for accurate threat prioritization
4. Creating an intuitive dashboard for security monitoring and management
5. Enabling automated blue team defense capabilities for proactive threat mitigation
6. Providing cost-benefit analysis tools for security investment decisions

The RTTI system represents a significant advancement in proactive security operations, enabling organizations to identify and respond to threats more effectively while optimizing security investments through data-driven decision making.

8. REFERENCES

NIST (2023). *Cybersecurity Framework 2.0*. National Institute of Standards and Technology. https://www.nist.gov/cyberframework

NIST (2022). *Special Publication 800-37 Rev. 2: Risk Management Framework for Information Systems and Organizations*. National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

MITRE (2024). *MITRE ATT&CK Framework*. The MITRE Corporation. https://attack.mitre.org/

Shodan (2024). *Shodan API Documentation*. https://developer.shodan.io/api
Hunt, T. (2024). *Have I Been Pwned API Documentation*. https://haveibeenpwned.com/API/v3
VirusTotal (2024). *VirusTotal API Documentation*. https://developers.virustotal.com/reference
Hugging Face (2025). *Hugging Face API Documentation*. https://huggingface.co/docs/api-inference/index
Supabase (2024). *Supabase Documentation*. https://supabase.com/docs
Vue.js (2025). *Vue.js 3 Documentation*. https://vuejs.org/guide/introduction.html
OWASP (2024). *OWASP Top Ten Web Application Security Risks*. Open Web Application Security Project. https://owasp.org/www-project-top-ten/
Morgan, S. (2024). "2024 Official Annual Cybercrime Report." *Cybersecurity Ventures*. https://cybersecurityventures.com/annual-cybercrime-report/
Cybersecurity & Infrastructure Security Agency (2024). *CISA Known Exploited Vulnerabilities Catalog*. CISA. https://www.cisa.gov/known-exploited-vulnerabilities-catalog
IBM Security (2024). *Cost of a Data Breach Report*. IBM & Ponemon Institute. https://www.ibm.com/security/data-breach
Node.js (2025). *Node.js Documentation*. https://nodejs.org/en/docs/
PostgreSQL (2024). *PostgreSQL Documentation*. https://www.postgresql.org/docs/
AWS (2024). *AWS EC2 Documentation*. Amazon Web Services. https://docs.aws.amazon.com/ec2/