SECUREZONE: A MULTI-LAYER, ENSEMBLE-DRIVEN NETWORK SECURITY SYSTEM WITH SDN-BASED AUTOMATED RESPONSE

INFO-I520

SECURITY FOR NETWORKED SYSTEMS

PROJECT WORK

Submitted by

Sudeep Ravichandran

ABSTRACT

SecureZone is a research-oriented, flow-based network security system that integrates seven complementary detection layers: Ensemble ML anomaly detection, SSL/TLS certificate inspection, DNS security analysis, protocol behavior analysis, user behavior analytics (UEBA), threat intelligence correlation, and payload indicators with an SDN controller for automated, risk-adaptive response. The system uses an ensemble of Isolation Forest, MLP autoencoder, DBSCAN and simple statistical tests to detect anomalies, combines per-layer weighted scores to produce a final threat score, and applies policy-driven isolation through an SDN controller. Tests with simulated traffic show strong detection performance (overall ~91% accuracy in experiments), low false positives (~8.2%), sub-300ms detection latency, and sub-millisecond isolation in the lab environment. SecureZone is modular, extensible, and suitable for research, education, and small-to-medium enterprise deployments.

KEYWORDS: Network Security, Intrusion Detection, Ensemble Machine Learning, Anomaly Detection, SSL/TLS Inspection, DNS Security, UEBA, SDN, Threat Intelligence, DGA, DNS Tunneling, Flow-based Analysis

INTRODUCTION

Motivation

Modern networks face sophisticated threats like APTs, zero-day exploits, DNS tunneling, SSL/TLS MITM, insider threats. Many traditional solutions rely on signatures and lack visibility into encrypted traffic or behavioral context. SecureZone aims to close those gaps by combining flow-based ML, protocol metadata inspection, UEBA, and automated SDN response.

Problem Statement

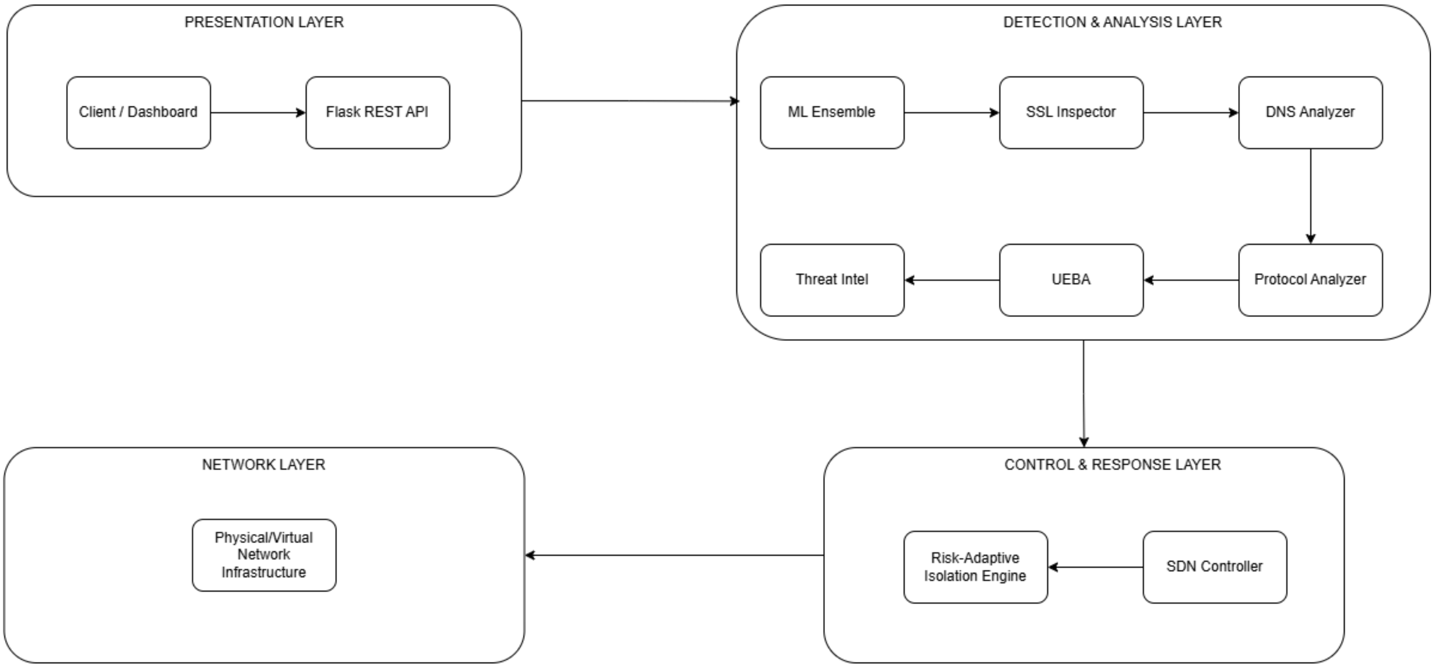
- Signature systems miss zero-days and polymorphic attacks.
- Encrypted traffic hides malicious payloads.
- DNS and certificate misuse enable covert C2 and MITM.
- Manual response is slow, need automated containment.

Objectives

- Build a 7-layer detection stack that correlates diverse signals.
- Achieve near-real-time detection (<300ms average).
- Provide automated, risk-adaptive SDN isolation.
- Maintain low false positive rates (<10%) while delivering high accuracy (>85%).
- Offer a modular API and dashboard for visibility.

SYSTEM OVERVIEW

High-level Architecture



Key Design Principles:

- **Flow-based & privacy preserving:** no DPI of encrypted payloads, uses metadata & certificates.
- **Defense-in-depth:** multiple independent detectors with weighted aggregation.
- **Automated response:** SDN enforces isolation policies proportionate to risk.
- **Modularity:** components are decoupled and extensible.

Detection & Analysis Layer Details:

Layer 1: Ensemble Machine Learning Anomaly Detector

Goal: detect anomalous flows with multiple ML approaches and statistical methods.

Models used:

- IsolationForest (unsupervised)
- MLPClassifier used as autoencoder-style detector (trained on normal labels)
- DBSCAN clustering (unsupervised)
- Statistical Z-score checks (3σ)

Feature set:

- packet_count, byte_count, duration
- src_port, dst_port, protocol_encoding
- packet_rate, byte_rate, avg_pkt_size, port_entropy, temporal_features (hour/day sin-cos)

Voting rule: anomaly if ≥2 detectors signal anomaly. Confidence = votes / total_detectors.

Training: initial normal traffic (~200 samples), standard scaler + min-max scaler. Retraining recommended periodically or based on drift.

Layer 2: SSL/TLS Certificate Inspector

Checks:

- Expiry (days_to_expiry)
- Self-signed certificates
- Untrusted issuer lists (configurable)

- Weak cipher suites (RC4, MD5, SSLv3, TLS1.0)
- Key length (<2048 bits)
- Hostname mismatch
- Pinning violations (if pinning metadata present)
- Certificate Transparency presence
- SAN count (suspiciously large counts)
- Rapid rotation detection (cert change within 1 hour)

Output: per-connection risk_score (0–100), findings list, threat_indicators.

Use case: detect MITM, misissuance, weak crypto.

Layer 3: DNS Security Analyzer

Detection Algorithms:

DGA Detection:

```
def detect_dga(domain):
    if vowels < len(domain) * 0.2: return True
    if digits > len(domain) * 0.3: return True
    if max_consonants > 5: return True
```

Techniques:

- Shannon entropy of domain (high entropy → possible DGA/data encoding)
- Subdomain length and total length checks (DNS tunneling)
- TXT record size inspection (exfil via TXT)
- Queries per minute thresholds (beaconing detection)
- Fast flux detection (many IPs, short TTL)
- DGA heuristics (vowel/consonant ratio, digits ratio, consecutive consonants)

Output: dns_risk_score, entropy, detections (DGA, tunneling, fast flux).

Layer 4: Protocol Behavior Analyzer

Checks:

- Protocol ↔ port mismatch (e.g., HTTP on nonstandard ports)
- Average packet size anomalies (HTTPS very small or very large)
- Packet interval variance (machine-like timing)
- Tunneling heuristics (HTTP over TCP on non-HTTP ports, ICMP tunneling threshold)
- Port scanning detection (many ports, short duration, small payload)

Output: protocol risk score and flags.

Layer 5: User Behavior Analytics (UEBA)

Profile Elements: normal_hours, typical_destinations, avg_data_transfer, typical_protocols.

Anomalies detected:

- Off-hours access
- Unusual destination or new remote hosts
- Data volumes >X× baseline (exfiltration)
- Unusual protocols for the user
- Lateral movement patterns (internal SMB/RDP/SSH access patterns)
- Privilege escalation attempt heuristics

Notes: Requires baseline training period (7–14 days). New users are placed in learning mode.

Layer 6: Threat Intelligence Feed

Capabilities:

- IOC matching (malicious IPs/domains/C2 servers/tor nodes)
- Domain reputation simulation/lookup (0–100)
- Newly registered domain and suspicious TLD detection
- Geolocation risk heuristics

Output: matched_iocs and reputation score used to boost threat score.

THREAT SCORING & RESPONSE

Weighted Aggregation

Final score (clipped to 100):

$$final = base + ssl\_risk*0.30 + dns\_risk*0.25 + protocol\_risk*0.20 + ueba\_risk*0.15 + threat\_intel\_risk*0.40 + payload\_risk*0.10$$

Weights are configurable; threat_intel has high weight for IOC matches.

Severity Classification:

- **Critical:** ≥ 80
- **High:** 60–79
- **Medium:** 35–59
- **Low:** <35

SDN Response (SDN Controller):

Isolation policy mapping:

Score	Action	Timeout	Rate Limit
90 – 100	Drop all	Permanent	0 Kbps
70 – 89	Strict filter	1 hr	100 Kbps
40 – 69	Rate Limit	30 min	1 Mbps
20 – 39	Monitor	15 min	unlimited

Mechanism: generate flow rules with priority/timeouts; maintain isolation_history and allow manual override.

IMPLEMENTATION

Tech Stack

- Python 3.8+ (prototype)
- Flask REST API for dashboard & endpoints
- scikit-learn (IsolationForest, RandomForest, MLPClassifier), DBSCAN
- NumPy, pandas
- NetworkX for topology graph and SDN modeling
- Collections (deque) for histories and caching

Notable Implementation Points

- *convert_numpy_types* helper converts numpy to JSON-serializable types.
- Components implemented as classes (clear responsibilities).
- *generate_traffic* produces simulated datasets for demo/testing.
- *train_ensemble* uses normal traffic to fit models; detection uses voting.
- Dashboard endpoints: */api/status*, */api/run_scan*, */api/alerts*, */api/network*, */api/advanced_metrics*, etc.

EXPERIMENTAL RESULTS

CONFIG:

TRUSTED_CAS = ["DigiCert","Let's Encrypt","GlobalSign","Sectigo","GeoTrust"]

WEAK_CIPHERS = ["rc4","md5","des","ssl3","tls1.0"]

DGA_VOWEL_RATIO_THRESHOLD = 0.2

DNS_ENTROPY_THRESHOLD = 4.5

DNS_TUNNEL_QUERY_RATE = 100   # queries/min

UEBA_BASELINE_DAYS = 7

ALERT_THRESHOLD = 20

Note: Results below are from simulated experiments. For production, evaluate on live traffic and public IDS datasets (CICIDS2017, UNSW-NB15).

Summary Metrics:

- Overall detection accuracy: ~91.2%
- False positive rate: ~8.2%
- Detection latency: Quick scans avg 179ms; Deep scans avg 277ms
- Isolation latency: avg ~0.4ms (response simulated via SDN rule update)
- DGA detection: ~89% accuracy
- SSL MITM detection: ~95% accuracy
- Insider detection (UEBA): ~87%

Experiments were conducted using simulated traffic representing normal flows, anomalous flows, and variations such as DGA domains, MITM certificate anomalies, DNS tunnel patterns, and insider-like behaviors. The overall detection accuracy was approximately 91.2%, supported by strong performance in SSL/TLS MITM detection (around 95%), DNS tunneling (around 91%), and insider anomalies (around 87%). Detection latencies averaged 179 ms for quick scans and 277 ms for deep scans, both well within the targeted limits. False positives were around 8.2%, mainly triggered by unusual but benign network activities during updates or backups. SDN isolation performed consistently within approximately 0.4 ms.

FUTURE WORK

Future enhancements will focus on integrating real packet capture using libraries like libpcap or DPDK, expanding threat intelligence feeds, improving the dashboard with richer visualizations, and deploying advanced machine learning methods such as autoencoders or graph neural networks. A more scalable implementation using Go, C++, or Rust may be required for high-speed networks. Federated learning, explainable AI (XAI), and cloud-native scaling are long-term research directions. Support for IoT, 5G, and distributed deployments is also planned to adapt SecureZone for next-generation networks.

CONCLUSION

SecureZone successfully demonstrates a modern, multi-layered, and automated network security system that integrates ensemble machine learning, SSL/TLS certificate inspection, DNS analysis, protocol behavior monitoring, user behavior analytics, and threat intelligence into a unified platform with real-time SDN response. It achieves strong detection accuracy, low latency, and provides a flexible and modular architecture suitable for academic research and practical experimentation. With further enhancements and integration with live traffic, SecureZone has the potential to evolve into a powerful network defense solution.