Zero Trust Architecture (ZTA) Implementation

CSDP 605 - Software Engineering
Final Exam Project Report

Sagar Patel

December 11, 2025

1 Introduction

This report documents the implementation of a Zero Trust Architecture (ZTA) system using Software-Defined Networking (SDN) with Mininet and the Ryu controller. The system implements three core security mechanisms: ICMP filtering, content filtering, and protocol filtering, with comprehensive traffic logging and behavior analysis capabilities.

2 System Architecture

2.1 Component Diagram

The system consists of multiple interconnected modules as shown in Figure 1.

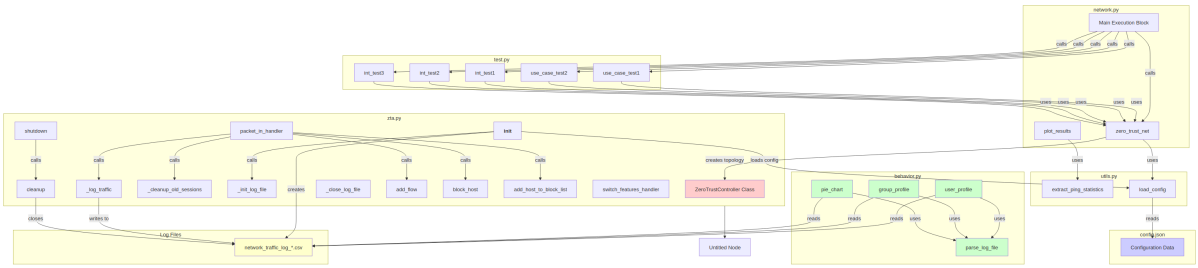


Figure 1: ZTA System Component Diagram showing the relationships between network.py, zta.py, behavior.py, test.py, and supporting modules.

Key components include:

- **network.py**: Main execution block with network topology creation
- **zta.py**: ZeroTrustController implementing security mechanisms
- **behavior.py**: User and group profiling with visualization
- **behavior_AI.py**: AI-generated behavior analysis
- **utils.py**: Configuration loading and log parsing utilities

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.

2.2 Packet Processing Flowchart

Figure 2 illustrates the sequential packet processing logic.

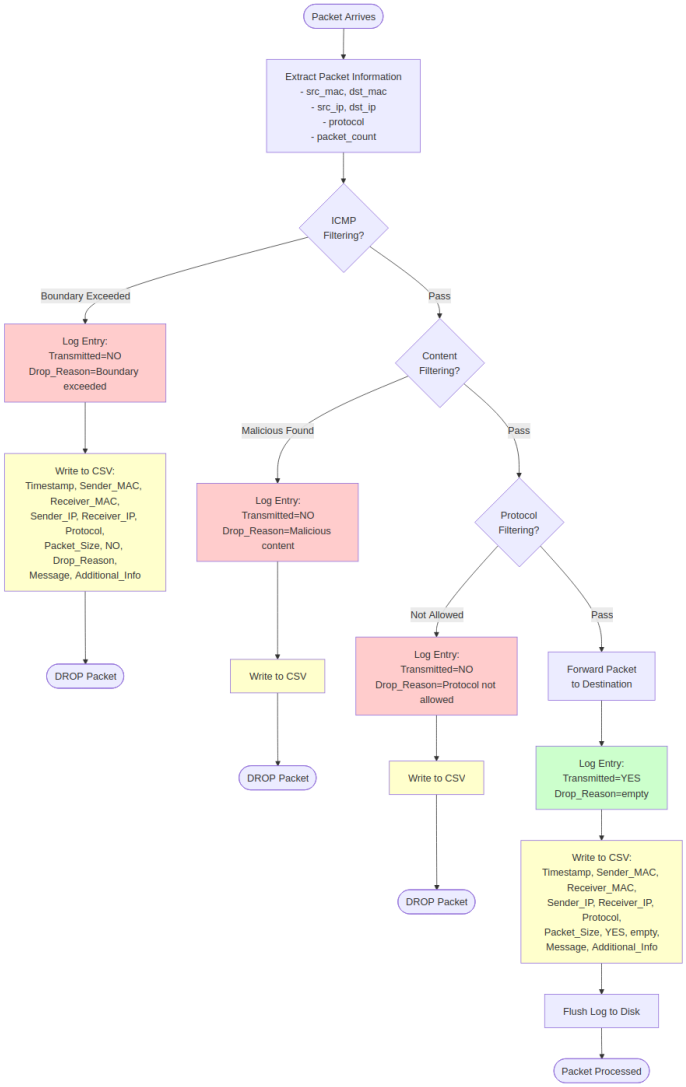


Figure 2: ZTA Packet Processing Flowchart showing ICMP, Content, and Protocol filtering stages.

Each packet passes through three filtering stages:

1. **ICMP Filtering**: Checks packet count against threshold (max_packets=11)
2. **Content Filtering**: Scans for malicious keywords
3. **Protocol Filtering**: Validates against allowed protocols (ICMP, TCP)

3 Security Mechanisms

3.1 ICMP Filtering

Monitors ICMP Echo Requests per session. When a host exceeds the configured threshold (11 packets within 200-second window), the host is added to the block list and subsequent packets are dropped.

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.

3.2 Content Filtering

Inspects packet payloads for malicious keywords including: “malware”, “attack”, “exploit”, “virus”, “trojan”, etc. Packets containing these keywords are logged and dropped.

3.3 Protocol Filtering

Maintains an allow-list of protocols (ICMP=1, TCP=6). Packets using unauthorized protocols are rejected.

4 Testing Results

4.1 Test 1: 5-Host Flood Attack Simulation

This test simulates a flood attack where Host 5 begins flooding Host 1 after 30 seconds of normal traffic.

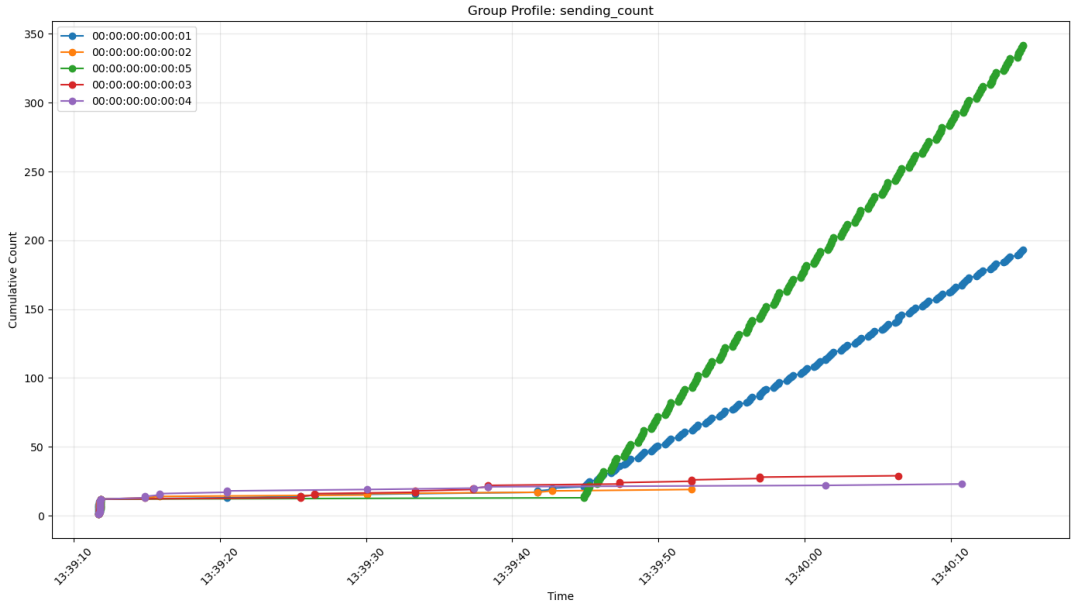
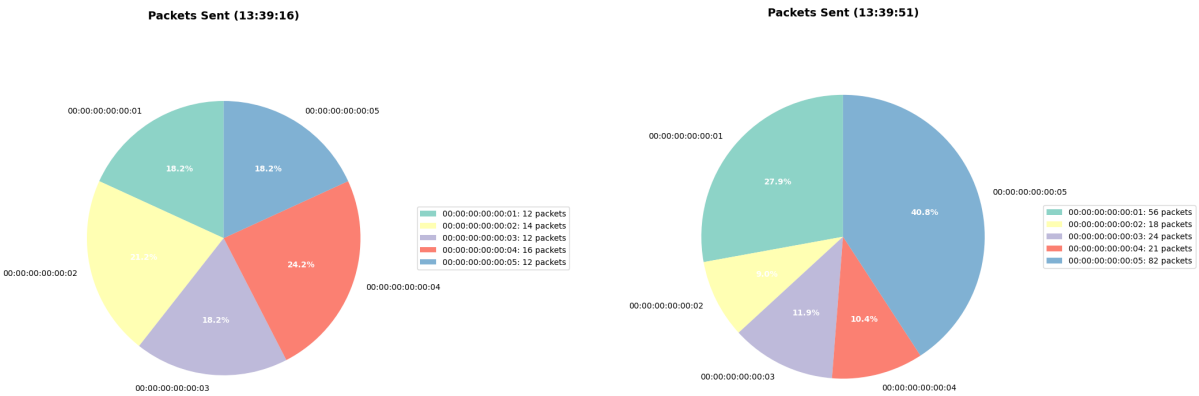


Figure 3: Test 1 - Group Profile: Packets Sent Over Time. Host 5 (h5) shows dramatic increase after 30 seconds.

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.



(a) At 5 seconds - Equal distribution (b) At 40 seconds - h5 dominates (40.8%)

Figure 4: Test 1 - Traffic Distribution Evolution showing flood attack progression

Observations:

- Initial phase (0-30s): All 5 hosts show similar traffic patterns (∼18% each)
- Flood phase (30-60s): Host 5 sends 330 packets (56.1% of total)
- Host 1 receives most packets as the flood target

4.2 Test 2: Aggressive Host Detection

This test validates the ICMP filtering mechanism with 3 hosts where Host 3 sends increasing packets.

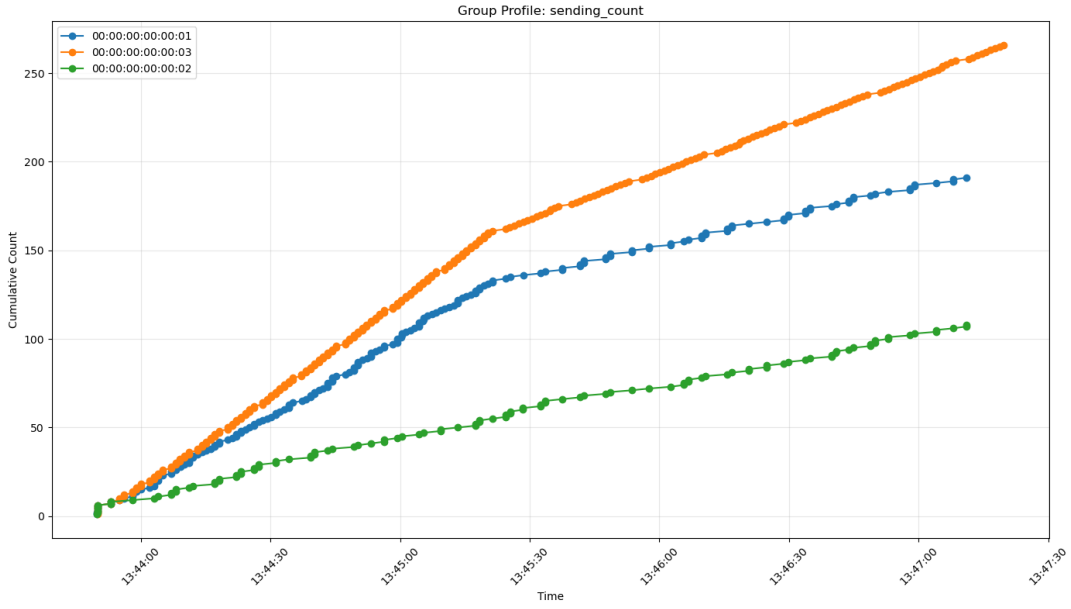
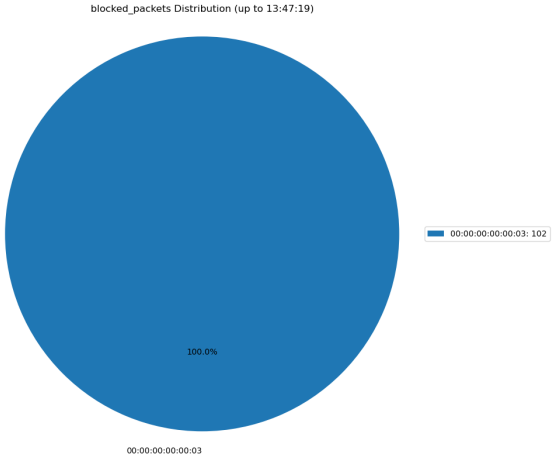
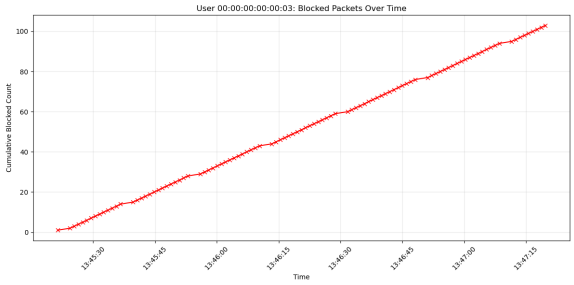


Figure 5: Test 2 - Group Profile: Packets Sent Over Time. Host 3 (green) shows highest sending rate.

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.



(a) Blocked packets - 100% from h3



(b) h3 blocked packets over time

Figure 6: Test 2 - Host 3 blocking results demonstrating ZTA effectiveness

Results:

- Host 3 sent 265 packets total
- 102 packets (38.5%) were blocked after threshold exceeded
- 100% of blocked packets came from h3 - confirming targeted blocking
- Hosts 1 and 2 continued normal operation unaffected

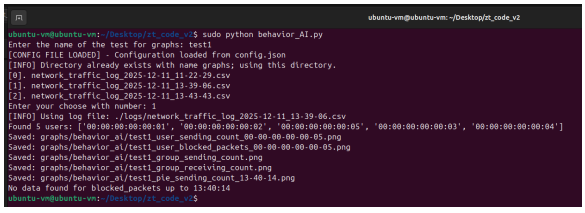
5 Behavior Analysis

The behavior analysis module provides four profiling methods:

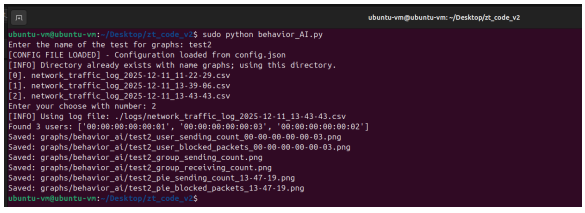
1. **sending_count**: Cumulative packets sent per user
2. **receiving_count**: Cumulative packets received per user
3. **blocked_packets**: Cumulative blocked packets per user
4. **protocol_distribution**: Transmitted packets by protocol (custom method)

5.1 AI-Generated Analysis

The `behavior_AI.py` module was executed on test logs to generate analysis graphs.



(a) Test 1 execution - 5 users detected



(b) Test 2 execution - 3 users detected

Figure 7: behavior_AI.py terminal output showing log processing

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.

6 Log File Summary

Three log files were generated during testing:

Log File	Test	Purpose
network_traffic_log_2025-12-11_11-22-29.csv	behavior/	Manual behavior analysis
network_traffic_log_2025-12-11_13-39-06.csv	test1	5-host flood simulation
network_traffic_log_2025-12-11_13-43-43.csv	test2	Aggressive host detection

Table 1: Log files generated during ZTA testing

Each log contains 11 fields: Timestamp, Sender_MAC, Receiver_MAC, Sender_IP, Receiver_IP, Protocol, Packet_Size, Transmitted, Drop_Reason, Message, Additional_Info.

7 Conclusion

The Zero Trust Architecture implementation successfully demonstrates:

1. **Effective ICMP Filtering**: Hosts exceeding packet thresholds are automatically blocked
2. **Real-time Monitoring**: All traffic is logged with comprehensive metadata
3. **Behavior Analysis**: User and group profiles enable anomaly detection
4. **Targeted Response**: Only malicious hosts are blocked; legitimate traffic continues

The system effectively identifies and mitigates flood attacks while maintaining network availability for legitimate users, demonstrating the core principles of Zero Trust security.

© 2025 Sagar Patel. All rights reserved. Do not copy or redistribute without permission.