Network Segmentation

# Solutions Architecture Document

**Version: 1.0**Date: October 10, 2025  
Classification: Internal

Prepared by: Network Security Team  
Auto-generated by Network Segmentation Analyzer

# Table of Contents

Note: After opening in Word, right-click here and select "Update Field" to generate the table of contents.

1. Executive Summary

2. Current State Analysis

2.1 Traffic Summary

2.2 Top Talkers

2.3 Suspicious Flows

3. Proposed Segmentation Design

3.1 Network Zones

3.2 Zone Architecture

3.3 Traffic Flow Patterns

4. Segmentation Rules

4.1 Rule Overview

4.2 Detailed Rules Table

4.3 Implementation Examples

5. Implementation Plan

5.1 Phased Rollout

5.2 Validation Tests

5.3 Success Criteria

6. Testing and Rollback Plan

7. ML-Based Predictions (for apps without traffic data)

7.1 Overview

7.2 Prediction Summary

7.3 Sample Predictions

7.4 Recommendations

8. Appendices

A. Raw Data Samples

B. Full Rules Export

C. Glossary

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.0 | 2025-10-10 | Network Security Team | Initial release - automated network segmentation analysis |

# Executive Summary

## Key Findings

* Analyzed 132 network flows across 3 applications
* Identified 12 suspicious flows requiring immediate attention
* Current network has 118 internal flows and 14 external connections
* Generated 18 segmentation rules to enhance security posture

## Recommendations

* Implement micro-segmentation across all application tiers to enforce least-privilege access
* Block management ports (SSH, RDP) from external networks immediately - HIGH PRIORITY
* Restrict database access to authorized application servers only
* Deploy zone-based firewall policies to control east-west traffic
* Implement continuous monitoring and logging for all cross-zone traffic
* Conduct phased rollout with validation testing at each stage

## Risk Summary

**Critical Risks:** 7 high-risk flows identified with scores above 70. These require immediate remediation to prevent potential security breaches.  
  
**Primary Risk Vectors:**

* External access to management ports
* Unrestricted database port exposure
* Lack of micro-segmentation between application tiers
* Suspicious scanner/bot activity detected

# Current State Analysis

## Traffic Summary

**Total Traffic Volume**

|  |  |
| --- | --- |
| Total Flows | 132 |
| Total Bytes | 0.03 GB |
| Total Packets | 18,302 |
| Internal Flows | 118 (89.4%) |
| External Flows | 14 (10.6%) |

### Protocol Distribution

|  |  |
| --- | --- |
| Protocol | Flow Count |
| tcp | 122 |
| udp | 8 |
| icmp | 2 |

## Top Talkers

### Top Sources (by bytes)

|  |  |
| --- | --- |
| Source IP | Bytes Transferred |
| 10.3.1.15 | 3.83 MB |
| 10.2.3.30 | 2.96 MB |
| 10.3.1.16 | 2.83 MB |
| 10.2.3.31 | 2.77 MB |
| 10.2.3.32 | 2.59 MB |
| 10.2.14.140 | 1.57 MB |
| 10.2.1.10 | 1.45 MB |
| 10.1.4.30 | 1.37 MB |
| 10.3.1.10 | 1.31 MB |
| 10.3.3.30 | 1.18 MB |

## Suspicious Flows

|  |  |  |  |
| --- | --- | --- | --- |
| Source | Destination | Risk Score | Reason |
| suspicious-host | web-srv-01 | 100 | External access; Management port 22 exposed |
| malicious-bot | jenkins-master | 100 | External access; Malicious hostname detected; Scanner/bot activity |
| external-scanner | lb-web-01 | 80 | External access; Scanner/bot activity |
| external-scanner | lb-web-01 | 80 | External access; Scanner/bot activity |
| suspicious-scan | analytics-api | 80 | External access |
| suspicious-scan | kibana-dashboard | 80 | External access |
| port-scanner | grafana-dashboard | 80 | External access; Scanner/bot activity |
| external-client | lb-web-01 | 30 | External access |
| external-client | lb-web-01 | 30 | External access |
| external-client | lb-web-01 | 30 | External access |

# Proposed Segmentation Design

The proposed micro-segmentation design implements a zero-trust architecture with clearly defined security zones and strict access controls between tiers.

## Network Zones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Zone Name | Type | Members | Security Level | Description |
| APP\_TIER | micro | 32 | 3 | Application servers |
| DATA\_TIER | micro | 14 | 4 | Databases and storage |
| DMZ | macro | 0 | 2 | Demilitarized zone for public services |
| EXTERNAL | macro | 0 | 1 | Internet-facing zone |
| INTERNAL | macro | 0 | 3 | Internal network |
| MANAGEMENT\_TIER | micro | 6 | 4 | Monitoring, orchestration, and management |
| MESSAGING\_TIER | micro | 3 | 3 | Message queues and event buses |
| WEB\_TIER | micro | 9 | 2 | Web servers and load balancers |

## Zone Architecture Diagram

**Note:** See outputs/diagrams/overall\_network.html for the interactive network diagram. To embed the diagram image here, convert the Mermaid .mmd file to PNG using:

mmdc -i overall\_network.mmd -o overall\_network.png

## Recommended Traffic Flow Patterns

* EXTERNAL → DMZ: HTTP/HTTPS traffic only
* DMZ → WEB\_TIER: Reverse proxy connections
* WEB\_TIER → APP\_TIER: Application protocol (port 8080/8443)
* APP\_TIER → DATA\_TIER: Database connections (MySQL, PostgreSQL)
* APP\_TIER → MESSAGING\_TIER: Message queue access (Kafka)
* APP\_TIER → CACHE\_TIER: Cache access (Redis)
* MANAGEMENT\_TIER → All Tiers: Monitoring and metrics collection
* All reverse traffic flows: DENIED (prevents data exfiltration)

# Segmentation Rules

## Rule Overview

**Total Rules Generated:** 18  
  
The segmentation rules implement a default-deny policy with explicit allow rules for legitimate traffic flows. Rules are prioritized based on security risk, with highest-risk denials at the top.  
  
**Rule Priority Scheme:**

* 100-199: Critical blocks (management ports from external)
* 200-299: Database protection rules
* 300-399: Public service access (HTTP/HTTPS)
* 400-799: Internal tier-to-tier communication
* 800-899: Management and monitoring
* 9999: Default deny all

## Detailed Rules Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Priority | Source | Destination | Protocol:Port | Action | Risk | Justification |
| 150 | DATA\_TIER | WEB\_TIER | any:any | DENY | 85 | Prevent reverse traffic from data tier to web tier - data exfiltration protection |
| 300 | any | WEB\_TIER | tcp:80 | ALLOW | 20 | Allow HTTP traffic to web tier - legitimate public access |
| 300 | any | WEB\_TIER | tcp:443 | ALLOW | 20 | Allow HTTPS traffic to web tier - legitimate public access |
| 400 | WEB\_TIER | APP\_TIER | tcp:8080,8443 | ALLOW | 15 | Allow web tier to communicate with application tier |
| 600 | APP\_TIER | DATA\_TIER | tcp:6379 | ALLOW | 20 | Allow app tier to access Redis cache |
| 700 | APP\_TIER | MESSAGING\_TIER | tcp:9092,5672 | ALLOW | 20 | Allow app tier to access message queues (Kafka, RabbitMQ) |
| 800 | MANAGEMENT\_TIER | any | tcp:9090,9100,4040 | ALLOW | 30 | Allow management tier to collect metrics from all tiers |
| 1101 | EXTERNAL | any | tcp:22 | DENY | 90 | Block SSH access from external networks - high security risk |
| 1102 | EXTERNAL | any | tcp:3389 | DENY | 90 | Block RDP access from external networks - high security risk |
| 1103 | EXTERNAL | any | tcp:23 | DENY | 90 | Block Telnet access from external networks - high security risk |
| 1204 | EXTERNAL | DATA\_TIER | tcp:3306 | DENY | 95 | Block direct MySQL access from external - data exfiltration risk |
| 1205 | EXTERNAL | DATA\_TIER | tcp:5432 | DENY | 95 | Block direct PostgreSQL access from external - data exfiltration risk |
| 1206 | EXTERNAL | DATA\_TIER | tcp:27017 | DENY | 95 | Block direct MongoDB access from external - data exfiltration risk |
| 1207 | EXTERNAL | DATA\_TIER | tcp:6379 | DENY | 95 | Block direct Redis access from external - data exfiltration risk |
| 1511 | APP\_TIER | DATA\_TIER | tcp:3306 | ALLOW | 25 | Allow app tier to access MySQL databases |
| 1512 | APP\_TIER | DATA\_TIER | tcp:5432 | ALLOW | 25 | Allow app tier to access PostgreSQL databases |
| 1513 | APP\_TIER | DATA\_TIER | tcp:27017 | ALLOW | 25 | Allow app tier to access MongoDB databases |
| 9999 | any | any | any:any | DENY | 0 | Default deny policy for zero-trust architecture |

## Implementation Examples

The rules can be implemented on various platforms. Examples are provided in the appendices and output files.

# Implementation Plan

## Phased Rollout Approach

### Phase 1: Preparation (Week 1-2)

* Review and validate segmentation rules
* Set up staging environment mirroring production
* Configure firewall infrastructure
* Install monitoring and logging tools
* Create rollback procedures

### Phase 2: Critical Security Rules (Week 3)

* Block management ports from external networks
* Restrict database port access
* Implement default-deny policy on perimeter
* Enable comprehensive logging
* Validate with penetration testing

### Phase 3: Micro-Segmentation (Week 4-5)

* Deploy zone-based internal firewalls
* Implement WEB\_TIER → APP\_TIER rules
* Implement APP\_TIER → DATA\_TIER rules
* Configure monitoring tier access
* Validate application functionality

### Phase 4: Full Deployment & Optimization (Week 6-8)

* Roll out to all applications
* Fine-tune rules based on production traffic
* Optimize performance and latency
* Conduct final security validation
* Document as-built configuration

## Success Criteria

* **Zero unauthorized cross-zone traffic**
* **All applications functioning normally**
* **Response time impact < 5%**
* **All monitoring dashboards operational**
* **Successful penetration test completion**
* **Documentation complete and approved**

# Testing and Rollback Plan

## Validation Tests

|  |  |  |
| --- | --- | --- |
| Test Name | Procedure | Expected Result |
| External Port Scan Test | Scan all external IPs for open management ports (22, 23, 3389) | All management ports should be filtered/closed |
| Database Access Test | Attempt database connections from non-APP\_TIER hosts | All connections should be blocked |
| Application Functionality Test | Execute full end-to-end application test suite | All tests pass with < 5% latency increase |
| Monitoring Connectivity Test | Verify monitoring tools can collect metrics from all zones | All metrics dashboards populated |

## Rollback Procedures

In the event of critical issues, execute the following rollback steps:

1. Step 1: STOP: Halt the rollout immediately
2. Step 2: Disable newly implemented firewall rules
3. Step 3: Restore backup firewall configuration
4. Step 4: Verify application functionality
5. Step 5: Document the failure reason and lessons learned
6. Step 6: Schedule retrospective meeting
7. Step 7: Revise implementation plan before retry

Rollback Time Objective: < 15 minutes for complete restoration of service

# Appendices

## Appendix A: Sample Network Flow Data

Sample network flow records analyzed during this assessment:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | Destination | Protocol | Port | Bytes | Risk |
| 10.1.2.15 | 10.1.3.20 | tcp | 8080 | 234 KB | Low |
| 10.1.3.20 | 10.1.4.30 | tcp | 3306 | 87 KB | Medium |
| 203.0.113.45 | 10.1.1.10 | tcp | 443 | 89 KB | Low |
| 185.220.101.23 | 10.1.2.15 | tcp | 22 | 456 B | High |
| 10.1.7.60 | 10.1.3.20 | tcp | 9090 | 1 KB | Low |

## Appendix B: Generated Output Files

|  |  |
| --- | --- |
| File Path | Description |
| outputs/segmentation\_rules.csv | Complete rules in CSV format |
| outputs/iptables\_rules.sh | Linux IPTables implementation |
| outputs/aws\_security\_groups.json | AWS Security Group definitions |
| outputs/diagrams/overall\_network.mmd | Overall network Mermaid diagram |
| outputs/diagrams/overall\_network.html | Interactive network diagram |
| outputs/diagrams/zone\_flows.mmd | Zone traffic flow diagram |
| outputs/analysis\_report.json | Complete analysis in JSON format |

## Appendix C: Glossary

|  |  |
| --- | --- |
| Term | Definition |
| ACL | Access Control List - network traffic filtering rules |
| DMZ | Demilitarized Zone - network segment for public-facing services |
| East-West Traffic | Network traffic between servers within the data center |
| Micro-segmentation | Granular security controls between workloads/applications |
| North-South Traffic | Network traffic entering/leaving the data center |
| NSG | Network Security Group - Azure firewall construct |
| Security Group | AWS EC2 instance-level firewall |
| Zero-Trust | Security model requiring verification for all access |