Understanding Hardware Programming for MSS Rules

Multi-Domain Segmentation Service (MSS) vs. Traditional IP ACLs

In modern enterprise network architectures, security policies can be enforced using traditional IP Access Control Lists (ACLs) or through advanced identity-based frameworks such as Multi-Domain Segmentation Service (MSS). The fundamental difference in how these two mechanisms program hardware TCAM (Ternary Content Addressable Memory) dictates network operational scalability.

- **Traditional IP ACL Programming:** Traditional ACLs evaluate policies directly against individual IP addresses, host prefixes, and subnet masks. Every distinct IP prefix or subnet block specified within a policy rule requires dedicated hardware lookup entries in switch TCAM.
- **MSS Label-Based TCAM Programming:** MSS decouples policy enforcement from underlying IP address topology by mapping endpoints to numerical security tags / labels (e.g., Tag 100 for IoT Devices, Tag 200 for IoT Servers). Instead of evaluating raw IP addresses, the switch hardware programs policies directly between security labels in TCAM.
- **Zero TCAM Expansion on Subnet Growth:** When new host IP addresses or subnets are added to a security group, MSS updates only the edge endpoint database (Longest Prefix Match / LPM lookup table). Because the TCAM policy rule evaluates static numerical labels (Source Tag 100 ⇨ Destination Tag 200), there is exactly zero increase in physical TCAM policy entries regardless of how many host IPs are added.
- **Elimination of Combinatorial Explosion:** With traditional IP ACLs, enforcing a full-mesh security policy between two groups containing multiple subnets results in exponential TCAM consumption (M source subnets × N destination subnets). For example, a full-mesh ACL between 128 IoT subnets and 64 server subnets consumes 8,192 TCAM entries (128 × 64), quickly exhausting switch memory. With MSS, a single TCAM entry enforcing rule coverage between the two labels is sufficient to protect the entire group.

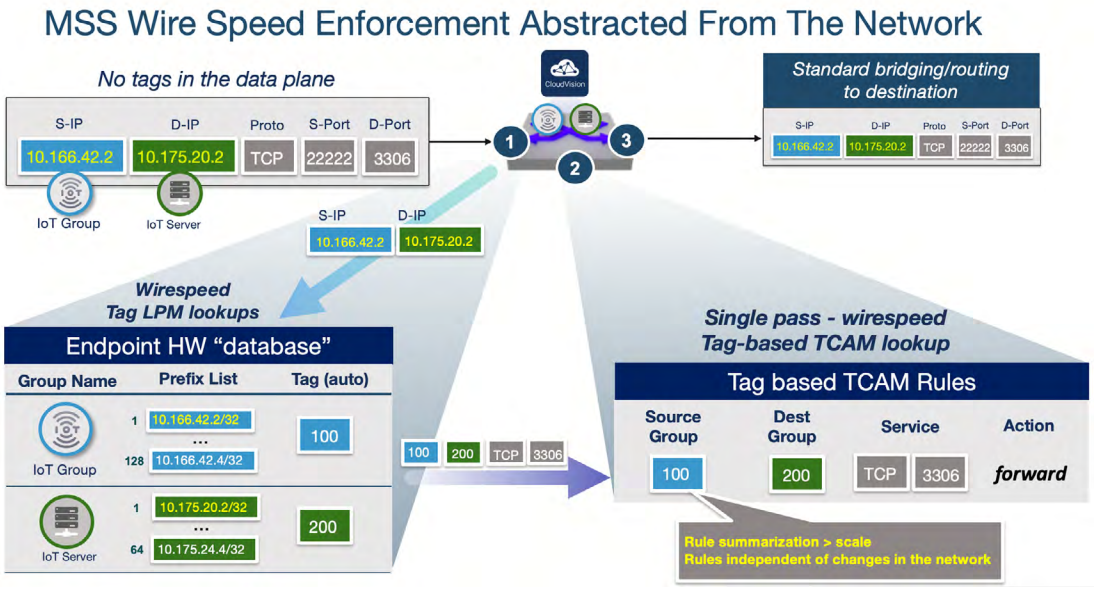


Figure 1: Multi-Domain Segmentation Service (MSS) Tag-Based TCAM Lookups

Hardware Label Programming Mechanics

The number of IPv4 prefixes that can be assigned to MSS groups is a function of the label scale for the platform in question. Let's examine how the hardware compiler programs internal labels for prefix objects during policy compilation.

Non-Overlapping Groups (1:1 Label Mapping)

When security groups share no common IP addresses, every prefix in Group G1s receives a uniform label, and every prefix in Group G10s receives a separate uniform label:

```
Non-Overlapping Group Label Allocation:
 Groups:
   G1s : 1.1.1.1/32, 2.2.2.2/32, 3.3.3.3/32
   G10s: 10.10.10.10/32, 20.20.20.20/32, 30.30.30.30/32
 Hardware Label Programming:
   1.1.1.1/32        --> 0x5, 0x0  (Group G1s)
   2.2.2.2/32        --> 0x5, 0x0  (Group G1s)
   3.3.3.3/32        --> 0x5, 0x0  (Group G1s)
   10.10.10.10/32 --> 0x4, 0x0  (Group G10s)
   20.20.20.20/32 --> 0x4, 0x0  (Group G10s)
   30.30.30.30/32 --> 0x4, 0x0  (Group G10s)
```

Overlapping Groups (Split Label Allocation)

When Group G-overlap is added, IP addresses 1.1.1.1/32 and 10.10.10.10/32 are present in multiple groups. Because a single label represents a specific security context, the hardware compiler must allocate distinct labels to shared prefixes versus unique prefixes to prevent classification ambiguity:

```
Overlapping Group Label Allocation:
 Groups:
   G-overlap: 1.1.1.1/32, 10.10.10.10/32, 11.11.11.11/32, 12.12.12.12/32
   G1s       : 1.1.1.1/32, 2.2.2.2/32, 3.3.3.3/32
   G10s      : 10.10.10.10/32, 20.20.20.20/32, 30.30.30.30/32
 Hardware Label Programming:
   1.1.1.1/32        --> 0x5, 0x0  (Shared in G-overlap & G1s)
   2.2.2.2/32        --> 0x8, 0x0  (Unique to G1s)
   3.3.3.3/32        --> 0x8, 0x0  (Unique to G1s)
   10.10.10.10/32 --> 0x4, 0x0  (Shared in G-overlap & G10s)
   11.11.11.11/32 --> 0x6, 0x0  (Unique to G-overlap)
   12.12.12.12/32 --> 0x6, 0x0  (Unique to G-overlap)
   20.20.20.20/32 --> 0x9, 0x0  (Unique to G10s)
   30.30.30.30/32 --> 0x9, 0x0  (Unique to G10s)
```

Overview of TCAM Expansion Factors

When traditional IP ACLs or stateless group policies are used, software security rules rarely map on a direct 1:1 basis to physical TCAM lookup entries. When security groups contain completely distinct, non-overlapping IP addresses, each software rule cleanly translates into a single hardware lookup entry (e.g., 2 rules yield 2 TCAM entries). However, when IP addresses overlap across multiple security groups, the hardware compiler must generate additional classification paths to resolve shared address memberships, causing a single software rule to expand into multiple hardware entries (such as 3 rules expanding into 5 TCAM entries).

Beyond IP address overlaps, hardware lookup footprints are further multiplied by two key policy dimensions: L4 service port specifications and rule directionality (where setting rules as bi-directional applies an exact 2× multiplier, expanding the 5 baseline TCAM entries into 10 physical hardware entries). Understanding these combined expansion vectors is critical for accurate hardware capacity planning and avoiding unexpected TCAM exhaustion.

Case Study

Non-Overlapping vs. Overlapping Address Group Behavior

Non-Overlapping Groups (1:1 Mapping)

When group definitions share no common IP addresses, each policy rule evaluates an entirely unique set of hosts. The hardware classification engine can program each rule as a single, contiguous TCAM table lookup entry.

Mapping Formula: N Policy Rules = N TCAM Entries (e.g., 2 non-overlapping rules result in 2 hardware entries).

Overlapping Groups (Hardware Path Expansion)

When an IP address belongs to multiple security groups, the hardware compiler must differentiate how packets matching that shared address are evaluated across different policy contexts. The compiler uses sophisticated masking logic to optimize the group entries (E.g Rule 1 below). However in certain worst case situations (e.g Rule 2 and 3 below) a single hardware entry cannot accommodate disjoint grouping logic without creating lookup ambiguity, the compiler splits the affected rules into separate, dedicated hardware labels.

To illustrate IP overlap expansion, consider a security policy with three configured address groups and three corresponding forwarding rules:

Configured Address Groups & IP Memberships:

- **Group G-overlap:** 1.1.1.1, 10.10.10.10, 11.11.11.11, 12.12.12.12
- **Group G1s:** 1.1.1.1, 2.2.2.2, 3.3.3.3
- **Group G10s:** 10.10.10.10, 20.20.20.20, 30.30.30.30

Identified Address Overlaps:

1. IP 1.1.1.1 is present in both G-overlap and G1s.
2. IP 10.10.10.10 is present in both G-overlap and G10s.

Configured Security Policy Rules:

- **Rule 1 (G-overlap):** Source = G-overlap | Destination = <any> | Service = SSH (TCP/22) | Action = Forward
- **Rule 2 (G1s):** Source = G1s | Destination = <any> | Service = SSH (TCP/22) | Action = Forward
- **Rule 3 (G10s):** Source = G10s | Destination = <any> | Service = SSH (TCP/22) | Action = Forward

When compiled into hardware, these 3 rules generate 5 distinct TCAM entries as verified in the switch CLI output:

CLI Output of TCAM Entries (3 Rules -> 5 Entries)

```
StrataAegis-Test-ipv4-in-253-1-1 uses 5 entries.

0x0000875e  0 hits - PmapVrfIntf0 Traffic Policy Rule: G-overlap-ip-0-tcp- dports: Value('Classification::PortRange', **
{'rangeStart': 22, 'rangeEnd': 22})

0x0000875f  0 hits - PmapVrfIntf0 Traffic Policy Rule: G1s-ip-0-tcp-      dports: Value('Classification::PortRange', **
{'rangeStart': 22, 'rangeEnd': 22})

0x00008760  0 hits - PmapVrfIntf0 Traffic Policy Rule: G1s-ip-0-tcp-      dports: Value('Classification::PortRange', **
{'rangeStart': 22, 'rangeEnd': 22})

0x00008761  0 hits - PmapVrfIntf0 Traffic Policy Rule: G10s-ip-0-tcp-     dports: Value('Classification::PortRange', **
{'rangeStart': 22, 'rangeEnd': 22})

0x00008762  0 hits - PmapVrfIntf0 Traffic Policy Rule: G10s-ip-0-tcp-     dports: Value('Classification::PortRange', **
{'rangeStart': 22, 'rangeEnd': 22})
```

Policy Rule	Source Group	Hardware Classification & Lookup Breakdown	TCAM Entries
Rule 1	G-overlap	All IPs in G-overlap (1.1.1.1, 10.10.10.10, 11.11.11.11, 12.12.12.12) compile into a single unified hardware lookup path.	1 Entry
Rule 2	G1s	Shared IP (1.1.1.1) and unshared IPs (2.2.2.2, 3.3.3.3) require split classification paths.	2 Entries
Rule 3	G10s	Shared IP (10.10.10.10) and unshared IPs (20.20.20.20, 30.30.30.30) require split classification paths.	2 Entries
Total	**3 Rules**	**Hardware compiler expands policy into 5 hardware lookup paths.**	**5 Entries**

Analysis

- **Rule 1 (G-overlap) ⇨ 1 TCAM Entry:** The IP addresses in group G-overlap (1.1.1.1, 10.10.10.10, 11.11.11.11, 12.12.12.12) compile together into a single primary hardware classification entry (0x0000875e).
- **Rule 2 (G1s) ⇨ 2 TCAM Entries:** Group G1s contains 1.1.1.1 alongside 2.2.2.2 and 3.3.3.3. Because 1.1.1.1 is also a member of G-overlap, the hardware compiler cannot evaluate 1.1.1.1 in the same lookup entry as 2.2.2.2 and 3.3.3.3 without causing lookup ambiguity. To maintain deterministic evaluation, Rule 2 is split into two hardware entries:
  - Entry 1 (0x0000875f): Dedicated to evaluating the shared IP 1.1.1.1.
  - Entry 2 (0x00008760): Dedicated to evaluating the unshared IPs 2.2.2.2 and 3.3.3.3.
- **Rule 3 (G10s) ⇨ 2 TCAM Entries:** Group G10s contains 10.10.10.10 alongside 20.20.20.20 and 30.30.30.30. Because 10.10.10.10 is also a member of G-overlap, the hardware compiler must isolate 10.10.10.10 from 20.20.20.20 and 30.30.30.30:
  - Entry 1 (0x00008761): Dedicated to evaluating the shared IP 10.10.10.10.
  - Entry 2 (0x00008762): Dedicated to evaluating the unshared IPs 20.20.20.20 and 30.30.30.30.

Non-Overlapping Group Case (1:1 Mapping)

If the policy contains no overlapping IP addresses (for example, evaluating only G1s and G10s without G-overlap), each rule translates 1-to-1 into a single TCAM entry:

CLI Output for Non-Overlapping Group TCAM Entries (2 Rules -> 2 Entries)

```
(Non-overlapping groups)

StrataAegis-Test-ipv4-in-253-1-0 uses 2 entries.

0x00008763  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-  dports:
Value('Classification::PortRange', ** {'rangeStart': 22, 'rangeEnd': 22})

0x00008764  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp- dports:
Value('Classification::PortRange', ** {'rangeStart': 22, 'rangeEnd': 22})
```

Impact of L4 Ports

L4 port specifications directly dictate how many hardware lookup conditions are required per IP classification path. The hardware compilation engine processes port definitions based on whether ports are continuous or non-continuous:

- **Single Port or Continuous Aligned Range:** A single port (e.g., TCP/22 for SSH) or a continuous range that aligns cleanly with hardware boundary blocks (e.g., TCP/49152-65535) can be evaluated within a single TCAM hardware entry per IP classification path.

CLI Output for Continuous Port Range TCAM entries (1 Port Block per IP Path)

```
# Continuous Port Range 49152-65535

StrataAegis-Test-ipv4-in-253-1-0 uses 5 entries.

0x000087a9  0 hits - PmapVrfIntf512 Traffic Policy Rule: G-overlap-ip-0-tcp- dports:
Value('Classification::PortRange', ** {'rangeStart': 49152, 'rangeEnd': 65535})

0x000087aa  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-      dports:
Value('Classification::PortRange', ** {'rangeStart': 49152, 'rangeEnd': 65535})

0x000087ab  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-      dports:
Value('Classification::PortRange', ** {'rangeStart': 49152, 'rangeEnd': 65535})

0x000087ac  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp-     dports:
Value('Classification::PortRange', ** {'rangeStart': 49152, 'rangeEnd': 65535})

0x000087ad  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp-     dports:
Value('Classification::PortRange', ** {'rangeStart': 49152, 'rangeEnd': 65535})
```

- **Non-Continuous Port Lists / Discrete Ports:** When a rule defines multiple discrete ports (e.g., SSH/22, HTTP/80, HTTPS/443), the hardware engine cannot evaluate them in a single pass. The rule must be split into separate sub-rules for each discrete port group.

Multiple Ports (5 Base Paths x 3 Ports = 15 Entries)

```
# Non-Overlapping Groups + 3 Discrete Ports (22, 80, 443):

StrataAegis-Test-ipv4-in-253-1-0 uses 6 entries. (2 groups x 3 ports = 6 entries)

0x000087bd  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-  dports: 22, 80, 443
0x000087be  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-  dports: 22, 80, 443
0x000087bf  0 hits - PmapVrfIntf512 Traffic Policy Rule: G1s-ip-0-tcp-  dports: 22, 80, 443
0x000087c0  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp- dports: 22, 80, 443
0x000087c1  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp- dports: 22, 80, 443
0x000087c2  0 hits - PmapVrfIntf512 Traffic Policy Rule: G10s-ip-0-tcp- dports: 22, 80, 443

# Overlapping Groups + 3 Discrete Ports (22, 80, 443):

StrataAegis-Test-ipv4-in-253-1-1 uses 15 entries. (5 base paths x 3 ports = 15 entries)
0x000087ae - 0x000087b0: G-overlap (3 entries)
0x000087b1 - 0x000087b6: G1s      (6 entries across 2 split paths)
0x000087b7 - 0x000087bc: G10s     (6 entries across 2 split paths)
```

Impact of Directionality

Rule directionality determines whether traffic classification is enforced in one direction or across both flow directions:

- **Uni-Directional Enforcement:** The TCAM programs entries solely for forward traffic matches (Source IP/Port ⇨ Destination IP/Port).
- **Bi-Directional Enforcement:** The policy compiler automatically generates a secondary, inverse classification entry (Destination IP/Port ⇨ Source IP/Port) to permit return flow traffic.
- **Exact 2x Hardware Multiplier:** Every baseline hardware entry generated by IP address overlaps and port range splits is duplicated for reverse direction matching.

CLI Output for Bi-Directional Rule Replication (Forward vs. Reverse Rules)

```
# Forward Path TCAM Entry Example:
0x00008781  0 hits - PmapVrfIntf0 Traffic Policy Rule: G-overlap-ip-0-tcp- dports: 80, 443, 8080, 8443

# Reverse Path Automatically Generated TCAM Entry Example:
0x00008785  0 hits - PmapVrfIntf0 Traffic Policy Rule: G-overlap-reverse-ip-0-tcp- sports: 80-80 443-443
8080-8080 8443-8443
```

Summary

Configuration Scenario	IP Overlap Factor	Port Specification	Directionality	TCAM Entries
Baseline (Non-Overlapping)	2 Rules (Distinct IPs)	Single Port (SSH/22)	Uni-Directional	2 Entries
Baseline Overlapping Case	3 Rules (Shared IPs)	Single Port (SSH/22)	Uni-Directional	5 Entries
Overlapping + Discrete Ports	3 Rules (Shared IPs)	3 Discrete Ports (22,80,443)	Uni-Directional	15 Entries (5×3)
Overlapping + Bi-Directional	3 Rules (Shared IPs)	Single Port (SSH/22)	Bi-Directional (2×)	10 Entries (5×2)
Full Multiplicative Scenario	**3 Rules (Shared IPs)**	**3 Discrete Ports (22,80,443)**	**Bi-Directional (2×)**	**30 Entries (5×3×2)**

Architectural Best Practices

- **Audit Address Group Overlaps:** Regularly scan security groups for shared IP addresses across policy rules and verify if they are required. Consolidating or isolating shared IPs prevents unexpected classification path splits.
- **Align Port Ranges:** Use continuous port range blocks aligned with hardware boundary registers where possible. Avoid defining long lists of non-continuous discrete ports.
- **Avoid Duplicate Rules:** If a rule is tagged as bi-directional, reverse rule is automatically programmed. Avoid creating forward and reverse rules and marking both as bidirectional. This will result in duplicate rules taking up TCAM entries
- **Factor Multiplicative Expansion in Capacity Planning:** When designing hardware ACLs, estimate capacity using the combined multiplicative formula rather than raw software rule counts.

Resources

Overview

MSS Overview

MSS Solution Brief

Zero Trust Maturity Model Whitepaper

MSS Technical Whitepaper

Datasheets

MSS for ZTN Datasheet

ZTX (MSS Traffic mapper) Datasheet

Design and Deployment Guides

MSS Campus Network and Security Design and Deployment Guide

ZTX Deployment Guide

MSS Deployment Guide

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062

Copyright © 2026 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. 06-0021-01 November 4, 2026