Enforcing Network-Wide Policies in the Presence of Dynamic Middlebox Actions using FlowTags

Seyed K. Fayazbakhsh*, Luis Chiang¶, Vyas Sekar*, Minlan Yu*, Jeffrey Mogul

*CMU, ¶Deutsche Telekom, *USC, ♦Google
Middleboxes complicate policy enforcement in SDN

Policy:
E.g., service chaining, access control

Dynamic and traffic-dependent modifications!
E.g., NATs, proxies
Modifications → Attribution is hard

Block the access of $H_2$ to certain websites.
Dynamic actions ➔ Policy violations

1. Get xyz.com
2. Response
3. Get xyz.com
4. Cached response

Proxy

Cached response

Web ACL
Block $H_2 \rightarrow xyz.com$

Internet
Our work: FlowTags

Some candidate (non-)solutions:
Placement, tunneling, consolidation, correlation

Address some symptoms but not root cause
→ OriginBinding and PathsFollowPolicy violations

FlowTags provides an architectural solution:
→ Enables policy enforcement and diagnosis despite dynamic middlebox actions.
Outline

• Motivation

• High-level Idea

• FlowTags Design

• Evaluation
High-level idea

• Middleboxes need to restore SDN tenets
  – Possibly only option for correctness
  – Minimal changes to middleboxes

• Add missing contextual information as Tags
  – NAT gives IP mappings,
  – Proxy provides cache hit/miss info

• FlowTags controller configures tagging logic
FlowTags architecture

New control apps
e.g., policy steering, verification

Network OS

Admin

Policy

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Control plane

Data plane

SDN Switches

FlowTable

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy

New control apps
e.g., policy steering, verification

Network OS

Existing APIs
e.g., OpenFlow

FlowTags APIs

FlowTags Tables

Mbox Config

FlowTags Enhanced Middleboxes

Admin

Policy
FlowTags in action

**Config w.r.t original principals**

**Block:** 10.1.1.2 $\rightarrow$ xyz.com

<table>
<thead>
<tr>
<th>&lt;SrcIP, Cache Hit&gt;</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.2, Hit</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>OrigSrcIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10.1.1.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Fwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$S_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Fwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ACL</td>
</tr>
</tbody>
</table>
Outline

• Motivation

• High-level Idea of FlowTags

• FlowTags Design

• Evaluation
Challenge 1: Tag Semantics

FlowTags-enhanced SDN Controller

Control plane

Data plane

Add Tag

Decode Tag

Proxy

Web ACL

H₁
10.1.1.1

S₁

S₂

H₂
10.1.1.2

Tag

Forward

Tag

Forward

Internet
Challenge 2: New APIs, control apps

FlowTags-enhanced SDN Controller

Control plane

Data plane

Add Tag | Decode Tag

Proxy | Web ACL

Tag Forward | Tag Forward

H₁ 10.1.1.1
H₂ 10.1.1.2
Challenge 3: Middlebox Extensions

FlowTags-enhanced SDN Controller

Control plane

Data plane

Add Tag

Decode Tag

Proxy

Web ACL

Internet

H₁
10.1.1.1

S₁
Tag Forward

S₂
Tag Forward

H₂
10.1.1.2
Outline

• Motivation

• High-level Idea of FlowTags

• FlowTags Design
  – Tag semantics
  – Controller and APIs
  – Middlebox modification

• Evaluation
Semantics: Dynamic Policy Graph (DPG)

Proxy

Web ACL: Block $H_2 \rightarrow xyz.com$

$H_1$

$H_2$

Proxy

ACL

Internet

$H_1$; Hit

$H_1$; Miss

$H_2$; Hit

$H_2$; Miss

$H_2$; <Allowed, Miss>

$H_2$; Blocked

$H_2$; <Allowed, Hit>

$H_1$; <Allowed, Hit>

$H_2$; Blocked

$H_2$; Miss

{H1}; Hit

{H1}; Miss

{H2}; Hit

{H2}; Miss

{H2}; <Allowed, Miss>

{H2}; Blocked

{H2}; Miss

{H1}; <Allowed, Hit>

{H1}; Miss

{H2}; Blocke

{H2}; Miss

{H2}; <Allowed, Hit>
Semantics: Dynamic Policy Graph (DPG)

Intuitively, need a Tag <per flow, per-edge> in DPG
Outline

• Motivation

• High-level Idea of FlowTags

• FlowTags Design
  – Tag semantics
  – Controller and APIs
  – Middlebox modification

• Evaluation
FlowTags-enhanced controller

Policy DPG

Physical realization

Middlebox Event Handlers

Switch Event Handlers

Reactive

Tag generate and consume

Flow expiry

Flow rules
Outline

• Motivation

• High-level Idea of FlowTags
  • FlowTags Design
    – Tag semantics
    – Controller and APIs
      – Middlebox modification

• Evaluation
Middlebox extension strategies to add FlowTags support

Strategy 1: Packet Rewriting

Pro: One shot
Con: Hard to get internal context
Middlebox extension strategies to add FlowTags support

Strategy 2: Module Modification

Pro: More change is needed
Con: Suited for getting internal context
Middlebox extension strategies to add FlowTags support

Our Strategy:
Packet rewriting for Tag consumption
Module modification for Tag generation
Outline

• Motivation

• High-level Idea of FlowTags

• FlowTags Design

• Evaluation
Key evaluation questions

• Feasibility of middlebox modification

• FlowTags overhead

• Number of Tag bits

• New capabilities
FlowTags needs minimal middlebox modifications

<table>
<thead>
<tr>
<th>Middlebox</th>
<th>Total LOC</th>
<th>Modified LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squid</td>
<td>216,000</td>
<td>75</td>
</tr>
<tr>
<td>Snort</td>
<td>336,000</td>
<td>45</td>
</tr>
<tr>
<td>Balance</td>
<td>2,000</td>
<td>60</td>
</tr>
<tr>
<td>iptables</td>
<td>42,000</td>
<td>55</td>
</tr>
<tr>
<td>PRADS</td>
<td>15,000</td>
<td>25</td>
</tr>
</tbody>
</table>
FlowTags adds low overhead

Breakdown of flow processing time (ms)

Controller Processing
Middlebox Tag Processing
Switch Setup

Abilene | Geant | Telstra | Sprint | Verizon | AT&T
---|---|---|---|---|---
11 | 22 | 44 | 52 | 70 | 115

# PoPs: 11 | 22 | 44 | 52 | 70 | 115
Summary of other results

• Adds < 1% overhead to middlebox processing

• Tags can be encoded in ~ 15 bits
  – E.g., IP-ID, IPv6 FlowLabel, EncapHeaders (NVP)

• Can enable new capabilities
  – Extended header space analysis
  – Diagnosing network bottlenecks
Conclusions

• Middleboxes complicate enforcement
  – E.g., NAT/LB rewrite headers, proxy sends cached response

• Root cause: Violation of the SDN tenets
  – Origin Binding and Paths-Follow-Policy

• FlowTags extends SDN with new middlebox APIs
  – Restores tenets using new DPG abstraction
  – No changes to switches and switch APIs

• FlowTags is practical
  – Minimal middlebox changes, low overhead
  – An enabler for verification, testing, and diagnosis