Class-Based Policy Provisioning:

Introducing Class-Based Policy Language (CPL)

August 2008
Class-Based Policy Provisioning

- Introduction
- Class-Based Policy Provisioning
- Class-Based Policy Language (CPL)
- Integrated Traffic Classification
- Configuring with CPL: Examples
- Monitoring and Statistics
- Roadmap
- Q&A
Many Features Act on Traffic

- Many features need to understand network traffic
  - Quality of Service
  - Security
  - Broadband
  - NetFlow
  - Routing
  - ... and many others

- Issue: Each feature might take a unique approach
  - Different configuration command syntax
  - Unnecessary complexity for customers
The Opportunity

- **Simplify**
  Simplify feature provisioning

- **Unify**
  Unify provisioning and behavior across platforms

- **Integrate**
  Make it easy to add new function and new platforms
Uniform Provisioning for Traffic Classification and Policy Actions

- Uniform Provisioning Across Features Across Platforms
- Unified Configuration Language
- Integrated Classification Definitions
- Greater Efficiency
CLASS-BASED POLICY PROVISIONING
Class-Based Policy Provisioning

A uniform, three step approach

- **Classification**
  Identify traffic of interest
  Specify match criteria that define a traffic class

- **Policy**
  Specify actions to take on the traffic class

- **Target**
  Apply the policy actions to a target
  Typically an interface or subinterface
Class-Based Policy: Terminology

- **What is a class of traffic?**
  A class is any traffic stream of interest
  Identify traffic streams by matching some criteria, such as
  - From a particular interface or port
  - Source or destination IP address
  - Protocol or application

- **What is a policy?**
  A policy is any action applied to a class
  Policies for Quality of Service, Security, Routing, Accounting, or Subscriber Service, such as
  - Assign higher or lower priority
  - Limit or drop traffic
  - Route on a different path

Example

Certain apples (the class) are selected for special handling (the policy)
Class-Based Policy: Terminology

- **What is a target?**
  
  A target defines a traffic stream to which a policy is applied

- Typically identifies the location, source or destination of traffic
  
  - Physical interfaces
    - Serial, Ethernet, POS
  
  - Logical interfaces
    - Subinterface, ATM VC, Frame Relay VC, VLAN
  
  - Logical entities
    - Control Plane Traffic
    - A Routing Protocol

  **Example**

  An orchard could be a source **(target)** to which a harvest policy is applied
Class-Based Policy Language (CPL)

- Unified method to specify classes, policies and targets
- Same framework for provisioning multiple features
  “Type” attributes for classes & policies

Benefit: Simpler provisioning
Across features
Across platforms
Specify Traffic Once, Take Multiple Actions

- With CPL features can share a *class-map*
  - Set up the classification criteria once
  - Use the *class-map* in different feature policies

- Benefits
  - Simplified configuration – policies point to same classification
  - Assured consistency – actions applied to same traffic

Example:

- Firewall policy permits selected traffic
- QoS policy assigns priority
Class-Based Policy Framework: Benefits

- Simpler for customers
  - Unified method for multiple features

- Faster time-to-market for new application recognition modules
  - Leverage by multiple features

- Easier to add new features
  - Leverage familiar provisioning method
    - Example (future): anomaly detection

- Quicker integration of new classification capabilities
  - Directly available to existing features
  - Enable new policy actions
INTEGRATED TRAFFIC CLASSIFICATION
Benefit: Integrated Classification Definitions

- Common definitions for protocol and application recognition
- Benefits:
  - Consistent classification results
  - New definitions available to all features

Shared definitions for all features:
- Protocols
- Applications
- Signatures
- QoS
- FPM
- Firewall
- NAT
- IPS
- NetFlow
- Routing
- ISA
Benefit: Dynamic Availability of New Definitions

- Dynamic addition of new definitions
  - Immediately available to all features
  - Live updates to in-service routers
  - Incorporate new definitions into live IOS images on the router

New definitions effective immediately

- QoS
- FPM
- Firewall
- NAT
- IPS
- NetFlow
- Routing
- ISA
Benefit: Greater Efficiency

- Performance improvement
  Consolidated classification for multiple features
  Not separate classification actions for each feature

- Benefits:
  Lower CPU consumption
  Greater throughput
CPL CONFIGURATION EXAMPLES
CPL Configuration Examples

- Quality of Service
- Flexible Packet Matching
- IOS Firewall
- Multi-Topology Routing
- IP SLAs
- NetFlow
- Summary Comparison:
  QoS, FPM, Firewall, and MTR
**CPL Configuration Example**

**QoS Giving Priority to Interactive Traffic**

Configure *class-maps* that classify Citrix ICA traffic by ICA tag

- **class-map** match-any Citrix-high-medium-low
  - match protocol citrix ica-tag “0”
  - match protocol citrix ica-tag “1”
  - match protocol citrix ica-tag “2”

- **class-map** Citrix-background
  - match protocol citrix ica-tag “3”

Create a *policy-map* that allocates bandwidth for traffic matched by *class-maps*

- **policy-map** Citrix-traffic
  - class Citrix-high-medium-low
    - bandwidth percent 20
  - class Citrix-background
    - bandwidth percent 5
    - police cir 128000
    - conform-action transmit
    - exceed-action drop

Assign the *policy-map* to a router interface with a *service-policy*

- interface serial 0/0
- service-policy output Citrix-traffic
## CPL Configuration Example

### FPM Used to Drop Slammer Worm

- **Classification**
  - Key word: `class-map`
  - Configure `class-maps` that classify Slammer worm
    ```
    class-map type stack ip-udp
    match field ip protocol eq 17 next udp
    class-map access-control slammer
    match class-map stack ip-udp
    match field udp dport eq 1434
    match start ip version offset 224 size 4 eq 0x04011010
    ```

- **Policy**
  - Key word: `policy-map`
  - Create a `policy-map` that drops traffic matched by the `class-map`
    ```
    policy-map type access-control policy-slammer
    class slammer
    drop
    ```

- **Target**
  - Key word: `service-policy`
  - Assign the `policy-map` to a router interface with a `service-policy`
    ```
    interface ethernet 1/0 service-policy type access-control input policy-slammer
    ```
CPL Configuration Example
IOS Firewall Blocks Instant Messaging

Classification
Key word: class-map

Configure *class-maps* to identify port-misuse and classify HTTP

- class-map type inspect http port-misuse-class
  - match port-misuse im
- class-map type inspect http-traffic-1
  - match protocol http

Policy
Key word: policy-map

Create *policy-maps* to terminate IM connections but permit desired HTTP traffic

- policy-map type inspect http myL7policy
  - class port-misuse-class
  - reset
- policy-map type inspect firewall-policy
  - class http-traffic-1
  - inspect

Target
Key word: service-policy

Assign the *policy-map* to a router interface with a *service-policy*

- interface pos 0/0
- service-policy type http myL7Policy
CPL Configuration Example
Multi-Topology Routing

Video traffic routed separately from other traffic

Classification
Key word: class-map

```
class-map match-any STANDARD_CLASS
match ip dscp default
```

```
class-map match-any VIDEO_CLASS
match ip dscp af43
```

Policy
Key word: policy-map

```
policy-map type class-routing ipv4 unicast MTR_POLICY
```

```
class STANDARD_CLASS
select-topology STANDARD
```

```
class VIDEO_CLASS
select-topology VIDEO
```

Target
Key word: service-policy

```
global-address-family ipv4
```

```
topology STANDARD
```

```
topology VIDEO
```

```
service-policy type class-routing MTR_POLICY
```

Source: Configuration fragment, MTR demo 9/2005
CPL Configuration Example
IP SLAs Integrated with a QoS Policy

Using IP SLAs to monitor a traffic class

Classification
Key word: class-map

```
class-map match-any VOIP
match ip dscp EF
class-map match-any biz
match ip dscp AF41
class-map default
```

Policy
Key word: policy-map

```
policy-map high-priority
class VOIP
measure gold-sla
bandwidth 2000
```

Target
Key word: service-policy

```
interface serial 0/0
service-policy output high-priority
```

IP SLAs action within a policy-map
Key word: measure

IP SLAs configuration
## CPL Configuration Example
### NetFlow Input Filters

<table>
<thead>
<tr>
<th>Classification</th>
<th>Key word: class-map</th>
<th><code>class-map high_importance_class</code></th>
<th>match access-group 101</th>
<th>Defines traffic class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><code>flow-sampler-map high_sampling</code></td>
<td>mode random one-out-of 1</td>
<td>Defines a NetFlow sampler</td>
</tr>
<tr>
<td>Policy</td>
<td>Key word: policy-map</td>
<td><code>policy-map mypolicy</code></td>
<td>class high_importance_class</td>
<td>Includes NetFlow sampling action in policy</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>flow-sampler high_sampling</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>Key word: service-policy</td>
<td><code>interface POS1/0</code></td>
<td>service-policy input mypolicy</td>
<td>Applies policy with Netflow sampling action to interfaces</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>interface ATM2/0</code></td>
<td>service-policy input mypolicy</td>
<td></td>
</tr>
</tbody>
</table>

**Classification**
- Key word: `class-map`
- `class-map high_importance_class`
  - match access-group 101

**Policy**
- Key word: `policy-map`
- `policy-map mypolicy`
  - class high_importance_class
  - `flow-sampler high_sampling`

**Target**
- Key word: `service-policy`
- `interface POS1/0` service-policy input mypolicy
- `interface ATM2/0` service-policy input mypolicy
## CPL Configuration Comparisons: Classification, Policy, Target

<table>
<thead>
<tr>
<th>CPL Step key word</th>
<th>Quality of Service</th>
<th>Flexible Packet Matching</th>
<th>IOS Firewall</th>
<th>Multi-Topology Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td><strong>class-map [type [subtype]]</strong></td>
<td><strong>class-map match-any Citrix-high-medium-low</strong> match protocol citrix ica-tag “0” match protocol citrix ica-tag “1” match protocol citrix ica-tag “2” <strong>class-map Citrix-background</strong> match protocol citrix ica-tag “3”</td>
<td><strong>class-map type stack ip-udp</strong> match field ip protocol eq 17 next udp <strong>class-map access-control-slammer</strong> match class-map stack ip-udp match field udp dport eq 1434 match start ip version offset 224 size 4 eq 0x04011010</td>
<td><strong>class-map type inspect http port-misuse-class</strong> match port-misuse im <strong>class-map type inspect http-traffic-1</strong> match protocol http</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td><strong>policy-map [type [subtype]]</strong></td>
<td><strong>policy-map Citrix-traffic</strong> class Citrix-high-medium-low bandwidth percent 20 class Citrix-background bandwidth percent 5 police cir 128000 conform-action transmit exceed-action drop</td>
<td><strong>policy-map type access-control policy-slammer</strong> class slammer drop</td>
<td><strong>policy-map type inspect http myL7policy</strong> class port-misuse-class reset <strong>policy-map type inspect firewall-policy</strong> class http-traffic-1 inspect</td>
</tr>
<tr>
<td><strong>Target</strong></td>
<td><strong>service-policy [type [subtype]]</strong></td>
<td><strong>interface serial 0/0 service-policy output Citrix-traffic</strong></td>
<td><strong>interface ethernet 1/0 service-policy type access-control input policy-slammer</strong></td>
<td><strong>interface pos 0/0 service-policy type http myL7Policy</strong></td>
</tr>
</tbody>
</table>
CBQoS MIB Shadows QoS Configuration

- Modular QoS Command Line Interface (MQC) is Cisco’s configuration language for Quality of Service
  
  Uniform interface for common QoS model across hardware platforms

- CBQoS MIB provides read access to configuration and statistical information for MQC

- Same structure as MQC

![Diagram of Configuration Language and QoS Behavioral Model](image)
Feature MIBs and CBP MIB

- CPL-provisioned features will link to two MIBs
  - Their own feature-specific MIB
  - The CBP MIB
Transition to Class-Based Policy MIB

Mirrors
Feature Adoption of CPL

- A MIB often reflects a feature’s provisioning syntax
  - The CBP MIB reflects the class-based provisioning model of CPL
  - Information that is common to multiple features must be accessible through a common MIB

- Provisioning method and feature MIBs

1. Non-CPL-Provisioned: Use existing feature MIB
2. CPL-Provisioned: Use CBP MIB with feature-specific MIB

![Diagram showing provisioning methods and feature MIBs]
Quality of Service: Evolution to CPL from MQC

- MQC is a proper subset of CPL
- Existing MQC configurations are forward-compatible to CPL
- **Router does not distinguish between CPL and MQC**
  - Common statistics are counted in *both* CBQoS MIB and new CBP MIB
  - Use one (CBQoS) or the other (CBP & new QoS)
  - CBQoS MIB will be supported indefinitely
ROADMAP CLASS-BASED POLICY PROVISIONING
### Class-Based Policy Provisioning Roadmap – IOS Release 12.4T

#### Features Using CPL

<table>
<thead>
<tr>
<th>Release 12.4T</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>Pre-12.4T</td>
</tr>
<tr>
<td></td>
<td>Quality of Service (QoS)</td>
</tr>
<tr>
<td></td>
<td>Control Plane Protection (CPPr)</td>
</tr>
<tr>
<td>2nd</td>
<td>12.4(4)T 11/14/2005</td>
</tr>
<tr>
<td></td>
<td>Flexible Packet Matching (FPM)</td>
</tr>
<tr>
<td>3rd</td>
<td>02/2006</td>
</tr>
<tr>
<td></td>
<td>IOS Firewall</td>
</tr>
<tr>
<td>4th</td>
<td>05/2006</td>
</tr>
<tr>
<td></td>
<td>FPM with CPL-XML</td>
</tr>
<tr>
<td></td>
<td>IP SLAs</td>
</tr>
<tr>
<td>6th</td>
<td>2H 2006</td>
</tr>
<tr>
<td></td>
<td>NetFlow</td>
</tr>
<tr>
<td></td>
<td>Intrusion Prevention System (IPS)</td>
</tr>
</tbody>
</table>

#### Platform Support

| Routers    | 800 Series, 1700 Series, 1800 Series, 2691, 2600XM Series, 2800 Series, 3700 Series, 3800 Series, 7200 Series, 7301 |
QUESTIONS AND ANSWERS
Q&A – 1

- Will existing MQC configurations still work?
  Yes. MQC syntax is a proper subset of CPL. Existing MQC configurations are forward-compatible with CPL.

- Will there be a “type” keyword for QoS? Will existing MQC configurations convert to it?
  Not at this time. Future evolution will determine the need. For the foreseeable future, QoS (MQC) configurations remain untyped.

- Will there be a tool to convert old style CLI to CPL syntax?
  There are no plans for a syntax conversion tool. Future evolution will determine the need. It is possible that some features may have a higher need than others and feature-specific converters may emerge.
Q&A – 2

- What is the order of operations when multiple CPL-configured polices are on an interface?

  CPL provisioning does not affect order of operations in the feature path.