Modular Policy Framework
January 26, 2009
By Steve Means

A topic that tends to confuse even experienced network engineers is modular policy framework. The configurations look arcane and seem to have their own strange language that humans don't naturally speak. Was it purposefully designed to confound us, or just written by bearded OS programmers who type more than they speak? One thing is clear. You need to know this for most Cisco professional level or above certifications, so there is no avoiding it. You might as well learn to speak the language.

Modular policy framework has three basic components:

The class map- Identifies traffic
The policy map- Applies actions to the identifies traffic
The service policy- Applies the policy map to an interface or globally

So lets work through a simple problem to show how to use MPL. You're given the task: On your router, drop any bittorrent connections from your internal network 10.1.1.0/24 to anywhere.

First we need to define an access list that we'll use in the class map to identify the 10.1.1.0 network.

R1(config)# **access list 10 permit 10.1.1.0 0.0.0.255** (Match any traffic that starts with 10.1.1.x)

Now we'll use a class map to identify the traffic, in this case matching both our access list, and bittorrent traffic:

R1(config)# **class-map match-all torrentmap** (Create a class map named torrentmap that will match all defined criteria)
R1(config-cmap)# **match protocol bittorrent** (the traffic first has to be bittorrent)
R1(config-cmap)# **match access-group 10** (*AND* it must match our ACL since we did a class-map match-all)

Now we need to apply an action to the traffic we've identified:

R1(config)# **policy-map torrentblock**
R1(config-pmap)# **class torrentmap** (on traffic that matches the class map…)
R1(config-pmap-c)# **drop** (drop the packet)

All that remains is to apply the policy to an interface. In this case since we want to block the traffic to anywhere, we'll apply it to the interface that the 10.1.1.0/24 network resides on. Lets say that interface is fa0/0.

R1(config-if)# **service-policy input torrentblock** (apply the policy to the interface in the incoming direction)

And that's all there is to it; identify the traffic, apply actions to the traffic, apply the policy to an interface.

Article Source:
http://www.ccbootcamp.com/support-resources/resources/articles-by-ccbootcamp.html