WHITE PAPER

DEPLOYING CONTROL PLANE POLICING

This document will provide an overview of Cisco IOS® Control Plane Policing (CPP), as well as deployment recommendations and guidelines for this Cisco IOS Security Infrastructure feature. CPP is used to increase security on Cisco routers by protecting the Route Processor from unnecessary and often malicious traffic. It allows users to configure a Quality of Service (QoS) filter that manages the traffic flow of control plane packets to protect the control plane of Cisco routers and switches against reconnaissance and Denial of Service (DoS) attacks allowing the Control Plane (CP) to maintain packet forwarding and protocol states despite an attack or heavy load on the router or switch.

PROTECTING THE ROUTE PROCESSOR

A router can be logically divided into four functional components or planes:

1. Data Plane
2. Management Plane
3. Control Plane
4. Services Plane

The vast majority of traffic travels through the router via the data plane; however, the Route Processor must handle certain packets, such as routing updates, keepalives, and network management. This is often referred to as control and management plane traffic.

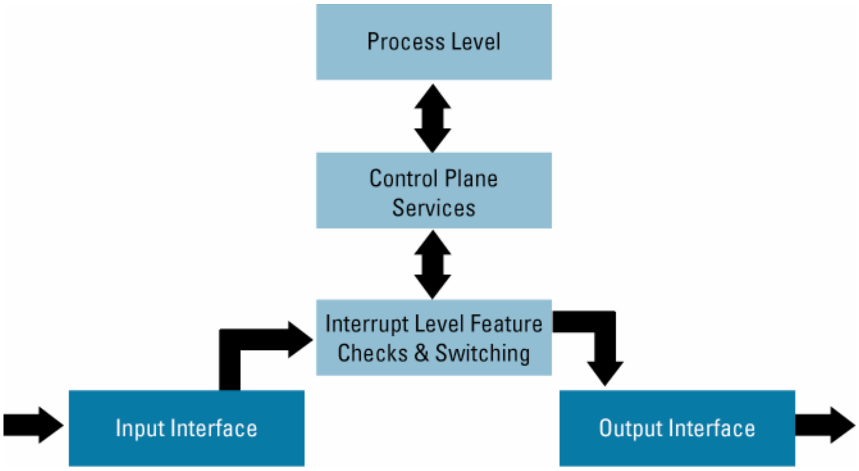
Because the Route Processor is critical to network operations, any service disruption to the Route Processor or the control and management planes can result in business-impacting network outages. A DoS attack targeting the Route Processor, which can be perpetrated either inadvertently or maliciously, typically involves high rates of punted traffic that result in excessive CPU utilization on the Route Processor itself. This type of attack, which can be devastating to network stability and availability, may display the following symptoms:

- High Route Processor CPU utilization (near 100%)
- Loss of line protocol keepalives and routing protocol updates, leading to route flaps and major network transitions
- Interactive sessions via the Command Line Interface (CLI) are slow or completely unresponsive due to high CPU utilization
- Route Processor resource exhaustion—resources such as memory and buffers are unavailable for legitimate IP data packets
- Packet queue backup, which leads to indiscriminate drops (or drops due to lack of buffer resources) of other incoming packets

CPP addresses the need to protect the control and management planes, ensuring routing stability, availability, and packet delivery. It uses a dedicated control-plane configuration via the Modular QoS CLI (MQC) to provide filtering and rate limiting capabilities for control plane packets.

Figure 1 illustrates the flow of packets from various interfaces. Packets destined to the control plane are subject to control plane policy checking, as depicted by the control plane services block.

Figure 1. Packet Flow



COMMAND SYNTAX

CPP leverages MQC to define traffic classification criteria and to specify configurable policy actions for the classified traffic. Traffic of interest must first be identified via class-maps, which are used to define packets for a particular traffic class. Once classified, enforceable policy actions for the identified traffic are created with policy-maps. The `control-plane` global command allows the CP service policies to be attached to control plane itself.

There are four steps required to configure CPP:

1. Define a packet classification criteria

```
router(config)#class-map <traffic_class_name>
router(config-cmap)#match <access-group | protocol* | ip prec | ip dscp>
```

2. Define a service policy

```
router(config)#policy-map <service_policy_name>
router(config-pmap)#class <traffic_class_name>
router(config-pmap-c)# police <cir | rate> conform-action <transmit | drop > exceed-action <transmit
| drop>
cir       Committed information rate (Bits per second)
rate      Specify policy rate in packets per second (pps)
```

3. Enter control-plane configuration mode

```
router(config)#control-plane
```

* When using the ‘match protocol’ classification criteria, ARP is the only protocol supported. All other protocols need an ACE entry for classification purposes.

4. Apply QoS policy

```
service-policy     {input | output} <service_policy_name>
input              Assign policy-map to the input of an interface
output**           Assign policy-map to the output of an interface
```

COMMAND REFERENCES

Please refer to the following links for more information on the control plane policing and QoS command syntax.

Control Plane Policing: http://www.cisco.com/en/US/products/sw/iosswrel/ps1838/products_feature_guide09186a00801afad4.html#1027184

QoS Command Reference: http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_command_reference_book09186a00801a7ec7.html

CISCO HARDWARE AND CISCO IOS SOFTWARE SUPPORT

Refer to Table 1 for Cisco hardware and Cisco IOS Software support.

Table 1. Cisco Hardware and Cisco IOS Software Support

Cisco Hardware	Cisco IOS Software Release
Cisco 12000 Series Router	Release 12.0(29)S
Cisco 7600 Series	Release 12.2(18)SXD1
Cisco 6500 Series	Release 12.2(18)SXD1
Cisco 7200 Series Cisco 7500 Series	Release 12.2(18)S
Cisco 1751 Router Cisco 2600/2600-XM Series Cisco 3700 Series Cisco 7200 Series	Release 12.3(4)T
Cisco 1800 Series Cisco 2800 Series	Release 12.3(8)T
Cisco 3800 Series	Release 12.3(11)T

DEVELOPING A CPP POLICY

Since CPP filters traffic destined to the Route Processor, it is critical to gain an adequate level of understanding about the legitimate traffic destined for the Route Processor prior to deployment. Configuring CPP policies without this knowledge may result in the blockage of critical traffic, with the potential for unintentionally provoking a DoS attack. In some networks, determining the exact traffic profile required for CPP policies might be difficult, so a careful staged approach should be taken to define these policies. Refer to the Deployment Guidelines section of this document for a recommended conservative methodology for deploying CPP using iterative ACL configurations to help identify and filter traffic.

** Although MQC can be leveraged to support outbound policies, this document focuses solely on input CPP since input CPP provides the most effective protection scenario. Output CPP is mainly used to suppress responses to input packets and does not limit response generation.

Traffic Classification

Prior to developing the actual CPP policy, administrators must identify required traffic and separate it into different classes. Multiple classification schemes can be used, but Cisco recommends a methodology that involves dividing traffic into distinct groups based on relative importance.

The following example uses nine different classes of traffic, thus providing a granular level of detail for real-world environments. Use this as a reference; however, note that the actual number and type of classes needed for a network may differ and should be selected based on local requirements, security policies, and a thorough analysis of the baseline traffic of the customer.

The nine traffic classes in this example were created with the following criteria:

1. **Border Gateway Protocol (BGP)**

- Traffic that is crucial to maintaining neighbor relationships for BGP routing protocol
- Examples: BGP keepalives and routing updates
- Maintaining BGP routing protocol is crucial to maintaining connectivity within a network or to a Service Provider
- Sites that do not run BGP will not need to use this class

2. **Interior Gateway Protocol (IGP)**

- Traffic that is crucial to maintaining IGP routing protocols
- Examples: Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP)
- Maintaining IGP routing protocols is crucial to maintaining connectivity within a network

3. **Management**

- Necessary, frequently used traffic that is required during day-to-day operations
- Examples: traffic used for remote network access, Cisco IOS Image upgrades and management (ie: telnet, Secure Shell (SSH), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP), Terminal Access Controller Access Control System (TACACS), Hypertext Transfer Protocol (HTTP), Trivial File Transfer Protocol (TFTP), and File Transfer Protocol (FTP)

4. **Reporting**

- Traffic used for generating network performance statistics for reporting
- Example: using Cisco IOS IP Service Level Agreements (SLAs) to generate ICMP with different DSCP settings in order to report on response times within different QoS data classes

5. **Monitoring**

- Traffic used for monitoring a router
- Traffic should be permitted but should never pose a risk to the router; with CPP, this traffic can be permitted but limited to a low rate
- Examples: ICMP echo request (ping), and traceroute

6. **Critical Applications**

- Critical application traffic that is specific and crucial to a particular customer environment
- Traffic included in this class should be tailored specifically to the required application requirements of the user (ie: one customer may use multicast, while another uses IPSec and/or Generic Routing Encapsulation [GRE])
- Examples: GRE, Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), Session Initiation Protocol (SIP), Data Link Switching (DLSw), Dynamic Host Configuration Protocol (DHCP), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), Protocol Independent Multicast (PIM), Multicast Traffic, and IPSec

7. Layer 2 Protocols

- Traffic used for Address Resolution Protocol (ARP)
- Excessive ARP packets can potentially monopolize Route Processor resources, starving other important processes; CPP can be used to rate limit ARP packets to prevent this
- Currently, ARP is the only Layer 2 protocol that can be specifically classified using the “match protocol” classification criteria

8. Undesirable

- Explicitly identifies “bad” or malicious traffic that should be unconditionally dropped and denied access to the Route Processor
- Particularly useful when known traffic destined for the router should always be denied and not placed into a default category; explicitly denying traffic allows the end-user to collect rough statistics on this traffic via show commands and offers some insight into the rate of denied traffic

9. Default

- All remaining traffic destined for the Route Processor that has not been identified
- MQC provides the default class, so the user can specify the treatment to be metered out to traffic not explicitly identified in the other user defined classes
- Give this traffic access to the Route Processor at a highly reduced rate
- With a default classification in place, statistics can be monitored to determine the rate of otherwise unidentified traffic destined to the control-plane. Once this traffic is identified, further analysis can be performed to classify it and if needed, the other CPP policy entries can be updated to account for this traffic

Classification Access Lists

Using the classification scheme defined in the previous section, commonly required traffic is identified with a series of Access Control Lists (ACLs). This example uses extended-named ACLs, which are recommended because of their flexibility in allowing the targeted removal and insertion of actions within the ACL. This enables classification ACLs to be updated without completely removing and re-adding them. Named ACLs can also be named using descriptive names to indicate their use.

In this example, the following named classification ACLs are used to classify the traffic into the recommended classes:

- coppacl-bgp: BGP traffic
- coppacl-igp: IGP traffic
- coppacl-management: management traffic
- coppacl-reporting: reporting traffic
- coppacl-monitoring: monitoring traffic
- copp acl-critical-app: critical application traffic
- coppacl-layer2: ARP traffic
- coppacl-undesirable: explicitly denies unwanted traffic (i.e. slammer worm in this example)

The ACLs will be used to build classes of traffic that are used to define the service policies.

Note: As CPP policies are applied to the control plane interface, only traffic destined for the Route Processor will be affected by the CPP policy. The destination key word ‘any’ can be used to classify all traffic destined to any interface on the router. The router can be further secured by specifying exact destination IP addresses, which limits the number of specific interfaces that can receive certain protocols.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.