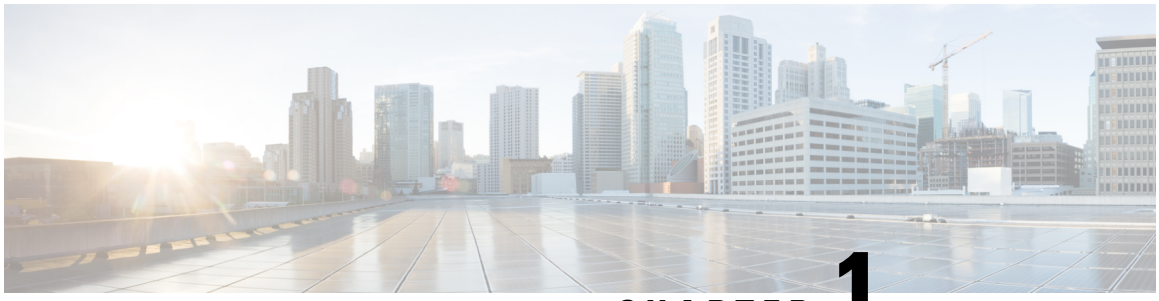
QoS: NBAR Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Classifying Network Traffic Using NBAR

Network-Based Application Recognition (NBAR) is a classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate quality of service (QoS) for that application or traffic with that protocol.

This module contains overview information about classifying network traffic in NBAR. The processes for configuring NBAR are documented in separate modules.

Note This module includes information for both NBAR and Distributed Network-Based Application Recognition (dNBAR). dNBAR is NBAR used on the Cisco 7500 router with a Versatile Interface Processor (VIP) and on the Catalyst 6500 family of switches with a FlexWAN module or serial interface processor (SIP). The implementation of NBAR and dNBAR is identical. Therefore, unless otherwise noted, the term NBAR is used throughout this module to describe both NBAR and dNBAR. The term dNBAR is used only when applicable.

Prerequisites for Using NBAR

Cisco Express Forwarding

Before you configure NBAR, you must enable Cisco Express Forwarding.

Note This prerequisite does not apply if you are using Cisco IOS Release 12.2(18)ZYA.

Stateful Switchover Support

NBAR is not supported with stateful switchover (SSO). This restriction applies to the Catalyst 6500 switches and to the Cisco 7500 and Cisco 7600 series routers.

Memory Requirements for dNBAR

To use dNBAR on a Cisco 7500 series router, you must be using a slot controller (or VIP processor) that has at least 64 MB of DRAM. Therefore, before configuring dNBAR on your Cisco 7500 series router, review the DRAM specifications for your particular slot controller or VIP processor.

Restrictions for Using NBAR

NBAR does not support the following:

- More than 24 concurrent URLs, hosts, or Multipurpose Internet Mail Extension (MIME) type matches.

Note For Cisco IOS Release 12.2(18)ZYA and Cisco IOS Release 15.1(2)T, the maximum number of concurrent URLs, hosts, or MIME type matches is 56.

- Matching beyond the first 400 bytes in a packet payload in Cisco IOS releases before Cisco IOS Release 12.3(7)T. In Cisco IOS Release 12.3(7)T, this restriction was removed, and NBAR now supports full payload inspection. The only exception is that NBAR can inspect custom protocol traffic for only 255 bytes into the payload.
- Non-IP traffic.
- Multiprotocol Label Switching (MPLS)-labeled packets. NBAR classifies IP packets only. You can, however, use NBAR to classify IP traffic before the traffic is handed over to MPLS. Use the modular quality of service (QoS) CLI (MQC) to set the IP differentiated services code point (DSCP) field on the NBAR-classified packets and make MPLS map the DSCP setting to the MPLS experimental (EXP) setting inside the MPLS header.
- Multicast and other non-Cisco Express Forwarding switching modes.
- Fragmented packets.
- Pipelined persistent HTTP requests.
- URL/host/MIME classification with secure HTTP.
- Asymmetric flows with stateful protocols.
- Packets that originate from or that are destined to the router running NBAR.

NBAR is not supported on the following logical interfaces:

- Fast Etherchannels

Note Fast Etherchannels *are* supported in Cisco IOS Release 12.2(18)ZYA.

- Dialer interfaces until Cisco IOS Release 12.2(4)T
- Interfaces where tunneling or encryption is used

Note You cannot use NBAR to classify output traffic on a WAN link where tunneling or encryption is used. Therefore, you should configure NBAR on other interfaces of the router (such as a LAN link) to perform input classification before the traffic is switched to the WAN link.

Note A Network Address Translation (NAT)-enabled Cisco device may experience an increase in CPU usage when upgrading the software from a previous release. Real Time Streaming Protocol (RTSP) and Media Gateway Control Protocol (MGCP) NAT Application Layer Gateway (ALG) support was added in Cisco IOS Release 12.3(7)T, which requires NBAR. Use the **no ip nat service nbar** command to disable NBAR processing, which can decrease the CPU utilization rate.

Warning If the **no ip nat service nbar** command is not specified during the startup of the router, results in the crashing of the router, when loading the configuration from the TFTP during the booting process.

Layer 2 NBAR Restrictions

The phrase "Layer 2 NBAR" refers to NBAR functionality used with Layer 2 interfaces (such as switchports, trunks, or Etherchannels).

Layer 2 NBAR functionality can be used with service modules such as a Firewall Service Module (FWSM) and an Intrusion Detection Service Module (IDSM) with the following restriction: Layer 2 NBAR is not supported on Layer 2 interfaces that are configured as part of a service module (such as FWSM and IDSM) when those service modules are configured in inline mode (that is, network traffic is in a direct path through the service module).

Note This restriction does not apply to NBAR functionality that is used with Layer 3 interfaces.

However, Layer 2 NBAR *is* supported in noninline mode with service modules even when Switched Port Analyzer (SPAN), Remote SPAN (RSPAN), or VLAN Access Control List (VACL) Capture functionality is used to send traffic to a service module.

For more information about the FWSM and its connection features, see the "Configuring Advanced Connection Features" module of the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide.*

For more information about the IDSM, see the "Configuring IDSM-2" module of the *Configuring the Cisco Intrusion Prevention System Sensor Using the Command Line Interface.*

For more information about SPAN or RSPAN, see the "Configuring SPAN and RSPAN" module of the *Catalyst 6500 Series Software Configuration Guide* .

For more information about VACL Capture, see the "VACL Capture for Granular Traffic Analysis with Cisco Catalyst 6000/6500 Running Cisco IOS Software" module.

Information About Classifying Network Traffic Using NBAR

NBAR Functionality

NBAR is a classification engine that recognizes and classifies a wide variety of protocols and applications, including web-based and other difficult-to-classify applications and protocols that use dynamic TCP/UDP port assignments.

When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate QoS for that application or traffic with that protocol. The QoS is applied using the modular quality of service CLI (MQC).

Note For more information about NBAR and its relationship with the MQC, see the "Configuring NBAR Using the MQC" module.

Examples of the QoS features that can be applied to the network traffic (using the MQC) after NBAR has recognized and classified the application or protocol include the following:

- Class-Based Marking
- Class-Based Weighted Fair Queuing (CBWFQ)
- Low Latency Queuing (LLQ)
- Traffic Policing
- Traffic Shaping

Note For Cisco IOS Release 12.2(18)ZYA on the Catalyst 6500 series switch (that is equipped with a Supervisor 32/programmable intelligent services accelerator [PISA]), only the following QoS features can be configured. These features can be configured (using the MQC) after NBAR has recognized and classified the application or protocol.

- Traffic Classification
- Traffic Marking
- Traffic Policing

Note For more information about the QoS features, see the "Quality of Service Overview" module. For more information about the Catalyst 6500 series switch and QoS, see the "Configuring QoS" module of the *Catalyst 6500 Series Software Configuration Guide.*

NBAR introduces several classification features that identify applications and protocols from Layer 4 through Layer 7. These classification features are as follows:

- Statically assigned TCP and UDP port numbers.
- Non-TCP and non-UDP IP protocols.
- Dynamically assigned TCP and UDP port numbers. This kind of classification requires stateful inspection; that is, the ability to inspect a protocol across multiple packets during packet classification.
- Subport classification or classification based on deep-packet inspection.

Deep-packet classification is classification performed at a finer level of granularity. For instance, if a packet is already classified as HTTP traffic, it may be further classified by HTTP traffic with a specific URL.

Note Access Control Lists (ACLs) can also be used for classifying static port protocols. However, NBAR is easier to configure and can provide classification statistics that are not available when ACLs are used.

NBAR includes a Protocol Discovery feature that provides an easy way to discover application protocols that are operating on an interface. For more information about Protocol Discovery, see the "Enabling Protocol Discovery" module.

Note NBAR classifies network traffic by application or protocol. Network traffic can be classified without using NBAR. For information about classifying network traffic without using NBAR, see the "Classifying Network Traffic" module.

NBAR Benefits

Identifying and classifying network traffic is an important first step in implementing QoS. A network administrator can more effectively implement QoS in a networking environment after identifying the number and types of applications and protocols that are running on a network.

NBAR gives network administrators the ability to see the different types of protocols and the amount of traffic generated by each protocol. After NBAR gathers this information, users can organize traffic into classes. These classes can then be used to provide different levels of service for network traffic, thereby allowing better network management by providing the appropriate level of the network resources for network traffic.

NBAR and Classification of HTTP Traffic

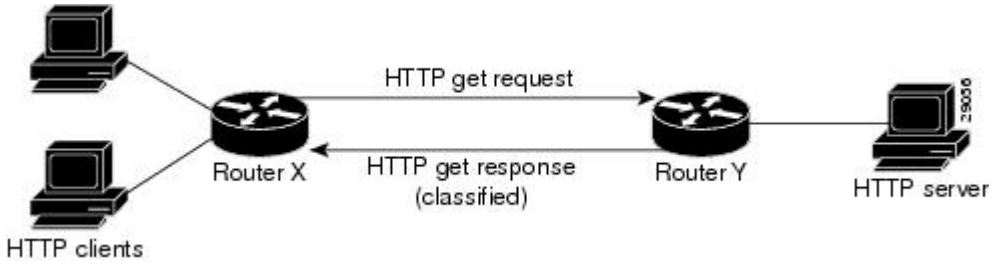
Classification of HTTP Traffic by URL Host or MIME

NBAR can classify application traffic by looking beyond the TCP/UDP port numbers of a packet. This is subport classification. NBAR looks into the TCP/UDP payload itself and classifies packets based on content within the payload such as that transaction identifier, message type, or other similar data.

Classification of HTTP traffic by URL, host, or Multipurpose Internet Mail Extension (MIME) type is an example of subport classification. NBAR classifies HTTP traffic by text within the URL or host fields of a request using regular expression matching. HTTP client request matching in NBAR supports most HTTP

request methods such as GET, PUT, HEAD, POST, DELETE, OPTIONS, CONNECT, and TRACE. The NBAR engine then converts the specified match string into a regular expression.

The figure below illustrates a network topology with NBAR in which Router Y is the NBAR-enabled router.



When specifying a URL for classification, include only the portion of the URL that follows the www.*hostname* .*domain* in the **match** statement. For example, for the URL www.cisco.com/latest/whatsnew.html, include only /latest/whatsnew.html with the **match** statement (for instance, **match protocol http url /latest/whatsnew.html**).

Note For Cisco IOS Release 12.2(18)ZY2 and later releases on the Cisco Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, and for Cisco IOS Release 15.1(2)T and later releases, up to 56 parameters or subclassifications per protocol per router can be specified with the **match protocol http** command. These parameters or subclassifications can be a combination of any of the available match choices, such as host matches, MIME matches, server matches, and URL matches. For other Cisco IOS releases and platforms, the maximum is 24 parameters or subclassifications per protocol per router.

Host specifications are identical to URL specifications. NBAR performs a regular expression match on the host field contents inside an HTTP packet and classifies all packets from that host. For example, for the URL www.cisco.com/latest/whatsnew.html, include only www.cisco.com.

For MIME type matching, the MIME type can contain any user-specified text string. A list of the Internet Assigned Numbers Authority (IANA) supported MIME types can be found at the following URL:

http://www.iana.org/assignments/media-types/

When matching by MIME type, NBAR matches a packet containing the MIME type and all subsequent packets until the next HTTP transaction.

NBAR supports URL and host classification in the presence of persistent HTTP. NBAR does not classify packets that are part of a pipelined request. With pipelined requests, multiple requests are pipelined to the server before previous requests are serviced. Pipelined requests are a less commonly used type of persistent HTTP request.

In Cisco IOS Release 12.3(4)T, the NBAR Extended Inspection for HTTP Traffic feature was introduced. This feature allows NBAR to scan TCP ports that are not well known and to identify HTTP traffic that traverses these ports. HTTP traffic classification is no longer limited to the well known and defined TCP ports.

Classification of HTTP Traffic Using HTTP Header Fields

In Cisco IOS Release 12.3(11)T, NBAR introduced expanded ability for users to classify HTTP traffic using information in the HTTP header fields.

HTTP works using a client/server model. HTTP clients open connections by sending a request message to an HTTP server. The HTTP server then returns a response message to the HTTP client (this response message

is typically the resource requested in the request message from the HTTP client). After delivering the response, the HTTP server closes the connection and the transaction is complete.

HTTP header fields are used to provide information about HTTP request and response messages. HTTP has numerous header fields. For additional information on HTTP headers, see section 14 of RFC 2616: Hypertext Transfer Protocol--HTTP/1.1. This RFC can be found at the following URL:

http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

NBAR can classify the following HTTP header fields:

- For request messages (client to server), the following HTTP header fields can be identified using NBAR:
  - User-Agent
  - Referer
  - From
- For response messages (server to client), the following HTTP header fields can be identified using NBAR:
  - Server
  - Location
  - Content-Encoding
  - Content-Base

Note Use of the Content-Base field has not been implemented by the HTTP community. (See RFC 2616 for details.) Therefore, the Content-Base field is not identified by NBAR on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Within NBAR, the **match protocol http c-header-field** command is used to specify that NBAR identify request messages (the "c" in the **c-header-field** portion of the command is for client). The **match protocol http s-header-field** command is used to specify response messages (the "s" in the **s-header-field**portion of the command is for server).

Note For Cisco IOS Release 12.2(18)ZY2 and later releases on the Cisco Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, and for Cisco IOS Release 15.1(2)T and later releases, the **c-header-field** and **s-header-field** keywords and associated arguments in the **match protocol http** command are not available. The same functionality is achieved by using the individual keywords and arguments. For more information, see the syntax of the **match protocol http** command in the Cisco IOS Quality of Service Solutions Command Reference.

Note The **c-header-field** performs subclassification based on a single value in the user agent, referrer, or from header field values and the **s-header-field** performs subclassification based on a single value that in the server, location, content-encoding, or content-base header field values. These header field values are not related to each other. Hence the **c-header** and **s-header** fields are replaced by user-agent, referrer, from, server, content-base, content-encoding, and location parameters as per the intent and need of HTTP subclassification.

Combinations of Classification of HTTP Headers and URL Host or MIME Type to Identify HTTP Traffic

Note that combinations of URL, Host, MIME type, and HTTP headers can be used during NBAR configuration. These combinations provide customers with more flexibility to classify specific HTTP traffic based on their network requirements.

NBAR and Classification of Citrix ICA Traffic

NBAR can classify Citrix Independent Computing Architecture (ICA) traffic and perform subport classification of Citrix traffic based on the published application name or ICA tag number.

Classification of Citrix ICA Traffic by Published Application Name

NBAR can monitor Citrix ICA client requests for a published application destined to a Citrix ICA Master browser. After the client requests the published application, the Citrix ICA Master browser directs the client to the server with the most available memory. The Citrix ICA client then connects to this Citrix ICA server for the application.

Note For Citrix to monitor and classify traffic by the published application name, Server Browser Mode on the Master browser must be used.

In Server Browser Mode, NBAR statefully tracks and monitors traffic and performs a regular expression search on the packet contents for the published application name specified by the **match protocol citrix** command. The published application name is specified by using the **app** keyword and the *application-name-string* argument of the **match protocol citrix** command. For more information about the **match protocol citrix** command, see the Cisco IOS Quality of Service Solutions Command Reference.

The Citrix ICA session triggered to carry the specified application is cached, and traffic is classified appropriately for the published application name.

Citrix ICA Client Modes

Citrix ICA clients can be configured in various modes. NBAR cannot distinguish among Citrix applications in all modes of operation. Therefore, network administrators might need to collaborate with Citrix administrators to ensure that NBAR properly classifies Citrix traffic.

A Citrix administrator can configure Citrix to publish Citrix applications individually or as the entire desktop. In the Published Desktop mode of operation, all applications within the published desktop of a client use the same TCP session. Therefore, differentiation among applications is impossible, and NBAR can be used to classify Citrix applications only as aggregates (by looking at port 1494).

The Published Application mode for Citrix ICA clients is recommended when you use NBAR. In Published Application mode, a Citrix administrator can configure a Citrix client in either seamless or nonseamless (windows) modes of operation. In nonseamless mode, each Citrix application uses a separate TCP connection, and NBAR can be used to provide interapplication differentiation based on the name of the published application.

Seamless mode clients can operate in one of two submodes: session sharing or nonsession sharing. In seamless session sharing mode, all clients share the same TCP connection, and NBAR cannot differentiate among applications. Seamless sharing mode is enabled by default in some software releases. In seamless nonsession

sharing mode, each application for each particular client uses a separate TCP connection. NBAR can provide interapplication differentiation in seamless nonsession sharing mode.

Note NBAR operates properly in Citrix ICA secure mode. Pipelined Citrix ICA client requests are not supported.

Classification of Citrix ICA Traffic by ICA Tag Number

Citrix uses one TCP session each time an application is opened. In the TCP session, a variety of Citrix traffic may be intermingled in the same session. For example, print traffic may be intermingled with interactive traffic, causing interruption and delay for a particular application. Most users likely would prefer that printing be handled as a background process and that printing not interfere with the processing of higher-priority traffic.

To accommodate this preference, the Citrix ICA protocol includes the ability to identify Citrix ICA traffic based on the ICA tag number of the packet. The ability to identify, tag, and prioritize Citrix ICA traffic is referred to as ICA Priority Packet Tagging. With ICA Priority Packet Tagging, Citrix ICA traffic is categorized as high, medium, low, and background, depending on the ICA tag of the packet.

When ICA traffic priority tag numbers are used, and the priority of the traffic is determined, QoS features can be implemented to determine how the traffic will be handled. For example, QoS traffic policing can be configured to transmit or drop packets with a specific priority.

Citrix ICA Packet Tagging

The Citrix ICA tag is included in the first two bytes of the Citrix ICA packet, after the initial negotiations are completed between Citrix client and server. These bytes are not compressed or encrypted.

The first two bytes of the packet (byte 1 and byte 2) contain the byte count and the ICA priority tag number. Byte 1 contains the low-order byte count, and the first two bits of byte 2 contain the priority tags. The other six bits contain the high-order byte count.

The ICA priority tag value can be a number from 0 to 3. The number indicates the packet priority, with 0 being the highest priority and 3 being the lowest priority.

To prioritize Citrix traffic by the ICA tag number of the packet, you specify the tag number using the **ica-tag** keyword and the *ica-tag-value* argument of the **match protocol citrix** command. For more information about the **match protocol citrix** command, see the Cisco IOS Quality of Service Solutions Command Reference.

NBAR and RTP Payload Type Classification

RTP is a packet format for multimedia data streams. It can be used for media-on-demand and for interactive services such as Internet telephony. RTP consists of a data and a control part. The control part is called Real-Time Transport Control Protocol (RTCP). RTCP is a separate protocol that is supported by NBAR. It is important to note that the NBAR RTP Payload Type Classification feature does not identify RTCP packets and that RTCP packets run on odd-numbered ports while RTP packets run on even-numbered ports.

The data part of RTP is a thin protocol that provides support for applications with real-time properties such as continuous media (audio and video), which includes timing reconstruction, loss detection, and security and content identification. RTP is discussed in RFC 1889 (*A Transport Protocol for Real-Time Applications*)and RFC 1890 (*RTP Profile for Audio and Video Conferences with Minimal Control*).

The RTP payload type is the data transported by RTP in a packet, for example, audio samples or compressed video data.

The RTP payload classification takes place in the persistent mode, wherein a fully qualified RTP session NBAR does the payload sub-classification. For example, RFC 2833 requires persistent processing for RTP payload sub-clasification within a classified flow.

The NBAR RTP Payload Type Classification feature not only allows real-time audio and video traffic to be statefully identified, but can also differentiate on the basis of audio and video codecs to provide more granular QoS. The RTP Payload Type Classification feature, therefore, looks deep into the RTP header to classify RTP packets.

NBAR and Classification of Custom Protocols and Applications

NBAR supports the use of custom protocols to identify custom applications. Custom protocols support static port-based protocols and applications that NBAR does not currently support. You can add to the set of protocols and application types that NBAR recognizes by creating custom protocols.

Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allows NBAR to classify nonsupported static port traffic.

Note For more information about specifying user-defined (custom) protocols, see the "Creating a Custom Protocol" module.

NBAR and Classification of Peer-to-Peer File-Sharing Applications

The following are the most common peer-to-peer file-sharing applications supported by NBAR:

- BitTorrent
- DirectConnect
- eDonkey
- eMule
- FastTrack
- Grokster
- JTella
- Kazaa (as well as Kazaa Lite and Kazaa Lite Resurrection)
- Morpheus
- Win MX

Gnutella Also Supported

The Gnutella file-sharing protocol became classifiable using NBAR in Cisco IOS Release 12.1(12c)E.

Applications that use the Gnutella protocol include Bearshare, Gnewtellium, Gnucleus, Gtk-Gnutella, Limewire, Mutella, Phex, Qtella, Swapper, and Xolo.

NBAR and Classification of Streaming Protocols

In Cisco IOS Release 12.3(7)T, NBAR introduced support for Real Time Streaming Protocol (RTSP). RTSP is the protocol used for applications with steaming audio, such as the following:

- Apple QuickTime
- RealAudio (RealSystems G2)
- Windows Media Services

NBAR and AutoQoS

In the earlier Cisco IOS releases the two features that allows to automate the deployment of QoS on your network: AutoQoS--Voice over IP (VoIP), and AutoQoS for the Enterprise. Both of these AutoQoS features take advantage of the traffic classification functionality of NBAR.

Note Cisco IOS Release 12.2(18)ZY (and later releases) does not support the AutoQoS--Voice over IP (VoIP) feature on the Catalyst 6500 series switch.

AutoQoS--VoIP

This feature was available with Cisco IOS Release 12.2(15)T. The AutoQoS--VoIP feature allows you to automate the deployment of QoS on your network and provides a means for simplifying the implementation and provisioning of QoS for VoIP traffic. For more information about the AutoQoS--VoIP feature and how it uses NBAR, see the "AutoQoS--VoIP" module.

AutoQoS for the Enterprise

This feature was available with Cisco IOS Release 12.3(11)T. The AutoQoS for the Enterprise feature allows you to automate the deployment of QoS in a general business environment, particularly for midsize companies and branch offices of larger companies. It expands on the functionality available with the AutoQoS--VoIP feature. For more information about the AutoQoS for the Enterprise feature and how it uses NBAR, see the "AutoQoS for the Enterprise" module.

NBAR and FWSM Integration

With Cisco IOS Release 12.2(18)ZYA, the functionality of NBAR to recognize protocols and applications was integrated with the Firewall Service Module (FWSM) on the Catalyst 6500 series switch. Available with this release were the following commands that can be used for classifying and tagging traffic to the FWSM:

- **ip nbar protocol-tagging**
- **show ip nbar protocol-tagging**

For more information about the FWSM and its connection features, see the "Configuring Advanced Connection Features" module of the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide.*

For more information about FWSM commands (including the two commands listed), see the Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Service Module Command Reference Guide.

NBAR and TelePresence PDLM

Cisco IOS Release 12.2(18)ZYA2 NBAR introduced support for the Cisco TelePresence PDLM.

Cisco TelePresence integrates advanced audio, high-definition video, and interactive elements to deliver an great meeting experience.

The Telepresence PDLM uses NBAR to identify TelePresence media and TelePresence control traffic over the network. TelePresence media traffic and TelePresence control traffic are treated differently by QoS and so must be classified separately. TelePresence media traffic must have a low latency. TelePresence control traffic does not require a low latency but should not be dropped.

NBAR-Supported Protocols

The **match protocol**(NBAR) command is used to classify traffic on the basis of protocols supported by NBAR. NBAR can classify the following types of protocols:

- Non-UDP and non-TCP IP protocols
- TCP and UDP protocols that use statically assigned port numbers
- TCP and UDP protocols that use statically assigned port numbers but still require stateful inspection
- TCP and UDP protocols that dynamically assign port numbers and therefore require stateful inspection

Many peer-to-peer file-sharing applications can be classified using FastTrack or Gnutella. See the NBAR and Classification of Peer-to-Peer File-Sharing Applications for additional information.

RTSP can be used to classify various types of applications that use streaming audio. See NBAR and Classification of Streaming Protocols for additional information.

The NBAR Protocol Pack provides an easy way to update protocols supported by NBAR without replacing the base IOS image that is already present in the device. A protocol pack is a set of protocols developed and packed together. For more information about loading an NBAR protocol pack, see the NBAR Protocol Pack module. To view the list of protocols supported in a protocol pack, see NBAR Protocol Library.

NBAR Memory Management

NBAR uses approximately 150 bytes of DRAM for each traffic flow that requires stateful inspection. (See NBAR Memory Management, on page 12 for a list of protocols supported by NBAR that require stateful inspection.)

When NBAR is configured, it allocates 1 MB of DRAM to support up to 5000 concurrent traffic flows. NBAR checks to see if more memory is required to handle additional concurrent stateful traffic flows. If such a need is detected, NBAR expands its memory usage in increments of 200 to 400 Kb.

Note This expansion of memory by NBAR does not apply if a PISA is in use.

NBAR Protocol Discovery

NBAR includes a feature called Protocol Discovery. Protocol discovery provides an easy way to discover the application protocols that are operating on an interface. For more information about protocol discovery, see the "Enabling Protocol Discovery" module.

Note With Cisco IOS Release 12.2(18)ZYA, which is intended for use on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA, Protocol Discovery supports Layer 2 Etherchannels.

Nonintrusive Protocol Discovery

Cisco IOS Release 12.2(18)ZYA1 includes the Nonintrusive Protocol Discovery feature which, enables the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA to perform protocol discovery in out-of-band (that is, offline) mode. In offline mode, a copy of the network traffic is used to discover the application protocols that are operating on an interface, leaving the network traffic undisturbed and available for other purposes.

Nonintrusive Protocol Discovery is closely associated with a feature called Intelligent Traffic Redirect (ITR). ITR allows network administrators to optimize system performance by identifying the specific traffic that needs to be redirected to the Supervisor 32/PISA for deep-packet inspection.

Nonintrusive Protocol Discovery is achieved by enabling ITR on an interface on which protocol discovery has been enabled. For more information about the commands used to enable ITR, see the Catalyst Supervisor Engine 32 PISA IOS Command Reference. For more information about protocol discovery, see the "Enabling Protocol Discovery" module.

Note For the Nonintrusive Protocol Discovery feature to function properly, no other "intrusive" features (for example, Flexible Packet Matching [FPM]) can be in use on the interface in either the input or output direction. An intrusive feature is one that somehow manipulates the packets (such as modifying a statistic or a packet counter). If such a feature is in use, the actual traffic (and not a copy of the traffic) is redirected.

NBAR Protocol Discovery MIB

The NBAR Protocol Discovery MIB expands the capabilities of NBAR Protocol Discovery by providing the following new functionality through Simple Network Management Protocol (SNMP):

- Enable or disable Protocol Discovery per interface.
- Display Protocol Discovery statistics.
- Configure and display multiple top-n tables that list protocols by bandwidth usage.
- Configure thresholds based on traffic of particular NBAR-supported protocols or applications that report breaches and send notifications when these thresholds are crossed.

For more information about the NBAR Protocol Discovery MIB, see the "Network-Based Application Recognition Protocol Discovery Management Information Base" module.

NBAR Categorization and Attributes

The NBAR Categorization and Attributes feature provides the mechanism to match protocols or applications based on certain attributes. As there are many protocols and applications, categorizing them into different groups will help with reporting as well as performing group actions, such as applying QoS policies, on them. Attributes are statically assigned to each protocol or application, and they are not dependent on the traffic. The following attributes are available to configure the match criteria using the **match protocol attribute** command. They are:

application-group: The **application-group** attribute allows the configuration of applications grouped together based on the same networking application as the match criteria. For example, Yahoo-Messenger, Yahoo-VoIP-messenger, and Yahoo-VoIP-over-SIP are grouped together under the yahoo-messenger-group.

category: The **category** attribute allows you to configure applications that are grouped together based on the first level of categorization for each protocol as the match criteria. Similar applications are grouped together under one category. For example, the email category contains all email applications such as, Internet Mail Access Protocol (IMAP), Simple Mail Transfer Protocol (SMTP), Lotus Notes, and so forth.

sub–category: The **sub–category** attribute provides the option to configure applications grouped together based on the second level of categorization for each protocol as the match criteria. For example, clearcase, dbase, rda, mysql and other database applications are grouped under the database group.

encrypted: The **encrypted** attribute provides the option to configure applications grouped together based on whether the protocol is an encrypted protocol or not as the match criteria. Applications are grouped together based on whether they are encrypted and non-encrypted status of the applications. Protocols for which the NBAR does not provide any value are categorized under the unassigned encrypted group.

tunnel: The **tunnel** attribute provides the option to configure protocols based on whether or not a protocol tunnels the traffic of other protocols. Protocols for which the NBAR does not provide any value are categorized under the unassigned tunnel group. For example, Layer 2 Tunneling Protocols (L2TP).

p2p-technology: The **p2p(Peer-to-Peer)-technology** attribute provides the option to indicate whether or not a protocol uses p2p technology.

Note Attribute-based protocol match configuration does not impact the granularity of classification either in reporting or in the protocol discovery information.

You can create custom values for the attributes application-group, category, and sub-category. The custom values enable you to name the attributes based on grouping of protocols. Use the **ip nbar attribute application-group custom application-group-name**, **ip nbar attribute category custom category-name**, and **ip nbar attribute sub-category custom sub-category-name** commands to add custom values for the attributes application-group, category, and sub-category, respectively.

The dynamically created custom attribute values can be used for attribute-map creation when using the **ip nbar attribute-map** command, and for configuring the match criterion for a class-map when using the **match protocol attribute** command.

The output from the **show ip nbar attribute-custom** command displays the number of custom values that can be defined for attributes, and the custom values that are currently defined. The **show ip nbar attribute** command displays all the attributes including the custom attributes used by NBAR.

To remove the custom values, use the **no ip nbar attribute** command.

NBAR Configuration Processes

Configuring NBAR consists of the following processes:

- Enabling Protocol Discovery (required)

When you configure NBAR, the first process is to enable Protocol Discovery.

- Configuring NBAR using the MQC (optional)

After you enable Protocol Discovery, you have the option to configure NBAR using the functionality of the MQC.

- Adding application recognition modules (also known as Packet Description Language Modules [PDLMs]) (optional)

Adding PDLMs extends the functionality of NBAR by enabling NBAR to recognize additional protocols on your network.

- Creating custom protocols (optional)

Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allow NBAR to classify nonsupported static port traffic.

NBAR Support for GETVPN

NBAR supports Group Encrypted Transport VPN (GETVPN). When ingress QoS is in crypto-map mode, the ingress QoS will work on encrypted traffic.

You can go back to backward compatible mode by using the **ip nbar disable classification encrypted-app** command in global configuration mode.

Note GETVPN is currently not supported by AVC and FNF.

Configuration Examples for Classifying Network Traffic Using NBAR

Example: Classification of HTTP Traffic Using the HTTP Header Fields

In the following example, any request message that contains "somebody@cisco.com" in the User-Agent, Referer, or From field will be classified by NBAR. Typically, a term with a format similar to "somebody@cisco.com" would be found in the From header field of the HTTP request message.

```
class-map match-all class1
 match protocol http from "somebody@cisco.com"
```

In the following example, any request message that contains "http://www.cisco.com/routers" in the User-Agent, Referer, or From field will be classified by NBAR. Typically, a term with a format similar to "http://www.cisco.com/routers" would be found in the Referer header field of the HTTP request message.

```
class-map match-all class2
 match protocol http referer "http://www.cisco.com/routers"
```

In the following example, any request message that contains "CERN-LineMode/2.15" in the User-Agent, Referer, or From header field will be classified by NBAR. Typically, a term with a format similar to "CERN-LineMode/2.15" would be found in the User-Agent header field of the HTTP request message.

```
class-map match-all class3
 match protocol http user-agent "CERN-LineMode/2.15"
```

In the following example, any response message that contains "CERN/3.0" in the Content-Base (if available), Content-Encoding, Location, or Server header field will be classified by NBAR. Typically, a term with a format similar to "CERN/3.0" would be found in the Server header field of the response message.

```
class-map match-all class4
 match protocol http server "CERN/3.0"
```

In the following example, any response message that contains "http://www.cisco.com/routers" in the Content-Base (if available), Content-Encoding, Location, or Server header field will be classified by NBAR. Typically, a term with a format similar to "http://www.cisco.com/routers" would be found in the Content-Base (if available) or Location header field of the response message.

```
class-map match-all class5
 match protocol http location "http://www.cisco.com/routers"
```

In the following example, any response message that contains "gzip" in the Content-Base (if available), Content-Encoding, Location, or Server header field will be classified by NBAR. Typically, the term "gzip" would be found in the Content-Encoding header field of the response message.

```
class-map match-all class6
 match protocol http content-encoding "gzip"
```

Example: Combinations of Classification of HTTP Headers and URL Host or MIME Type to Identify HTTP Traffic

In the following example, HTTP header fields are combined with a URL to classify traffic. In this example, traffic with a User-Agent field of “CERN-LineMode/3.0” and a Server field of “CERN/3.0”, along with host name “cisco.com” and URL “/routers”, are classified using NBAR:

```
Device(config)# class-map match-all c-http
Device(config-cmap)# match protocol http user-agent “CERN-LineMode/3.0”
Device(config-cmap)# match protocol http server "CERN/3.0"
Device(config-cmap)# match protocol http host cisco*
Device(config-cmap)# match protocol http url /routers
```

Example NBAR and Classification of Peer-to-Peer File-Sharing Applications

The **matchprotocolgnutellafile-transfer***regular-expression* and **matchprotocolfasttrackfile-transfer***regular-expression* commands are used to enable Gnutella and FastTrack classification in a traffic class. The **file-transfer** keyword indicates that a regular expression variable will be used to identify specific Gnutella or FastTrack traffic. The *regular-expression* variable can be expressed as "*" to indicate that all FastTrack or Gnutella traffic be classified by a traffic class.

In the following example, all FastTrack traffic is classified into class map nbar:

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*"
```

Similarly, all Gnutella traffic is classified into class map nbar in the following example:

```
class-map match-all nbar
 match protocol gnutella file-transfer "*"
```

Wildcard characters in a regular expression can also be used to identify specified Gnutella and FastTrack traffic. These regular expression matches can be used to match on the basis of a filename extension or a particular string in a filename.

In the following example, all Gnutella files that have the .mpeg extension will be classified into class map nbar.

```
class-map match-all nbar
 match protocol gnutella file-transfer "*.mpeg"
```

In the following example, only Gnutella traffic that contains the characters "cisco" is classified:

```
class-map match-all nbar
 match protocol gnutella file-transfer "*cisco*"
```

The following commands can be used for FastTrack traffic:

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*.mpeg"
```

or

```
class-map match-all nbar
 match protocol fasttrack file-transfer "*cisco*"
```

Example: NBAR and Classification of Custom Protocols and Applications

In the following example, the custom protocol app-sales1 will identify TCP packets that have a source port of 4567 and that contain the term "SALES" in the first payload packet:

```
Router(config)# ip nbar custom app-sales1 5 ascii SALES source tcp 4567
```

In the following example, the custom protocol virus-home will identify UDP packets that have a destination port of 3000 and that contain "0x56" in the seventh byte of the first packet of the flow:

```
Router(config)# ip nbar custom virus-home 7 hex 0x56 destination udp 3000
```

In the following example, the custom protocol media_new will identify TCP packets that have a destination or source port of 4500 and that have a value of 90 at the sixth byte of the payload. Only the first packet of the flow is checked for the value 90 at the offset 6.

```
Router(config)# ip nbar custom media_new 6 decimal 90 tcp 4500
```

In the following example, the custom protocol msn1 will look for TCP packets that have a destination or source port of 6700:

```
Router(config)# ip nbar custom msn1 tcp 6700
```

In the following example, the custom protocol mail_x will look for UDP packets that have a destination port of 8202:

```
Router(config)# ip nbar custom mail_x destination udp 8202
```

In the following example, the custom protocol mail_y will look for UDP packets that have destination ports between 3000 and 4000 inclusive:

```
Router(config)# ip nbar custom mail_y destination udp range 3000 4000
```

Example: Configuring Attribute-Based Protocol Match

The **match protocol attributes** command is used to configure different attributes as the match criteria for application recognition.

In the following example, the email-related applications category is configured as the match criterion:

```
Router# configure terminal
Router(config)# class-map mygroup
Router(config-cmap)# match protocol attribute category email
```

In the following example, skype-group applications are configured as the match criterion:

```
Router# configure terminal
Router(config)# class-map apps
Router(config-cmap)# match protocol attribute application-group skype-group
```

In the following example, encrypted applications are configured as the match criterion:

```
Router# configure terminal
Router(config)# class-map my-class
Router(config-cmap)# match protocol encrypted encrypted-yes
```

In the following example, Client-server sub-category applications are configured as the match criterion:

```
Router# configure terminal
RRouter(config)# class-map newmap
Router(config-cmap)# match protocol attribute sub-category client-server
```

In the following example, tunneled applications are configured as the match criterion:

```
Router# configure terminal
Router(config)# class-map mygroup
Router(config-cmap)# match protocol attribute tunnel tunnel-yes
```

The following sample output from the **show ip nbar attribute** command displays the details of all the attributes:

```
Router# show ip nbar attribute
      Name :  category
      Help :  category attribute
      Type :  group
    Groups :  email, newsgroup, location-based-services, instant-messaging, netg
      Need :  Mandatory
   Default :  other

      Name :  sub-category
      Help :  sub-category attribute
      Type :  group
    Groups :  routing-protocol, terminal, epayement, remote-access-terminal, nen
      Need :  Mandatory
   Default :  other

      Name :  application-group
      Help :  application-group attribute
      Type :  group
    Groups :  skype-group, wap-group, pop3-group, kerberos-group, tftp-group, bp
      Need :  Mandatory
   Default :  other

      Name :  tunnel
      Help :  Tunnelled applications
      Type :  group
    Groups :  tunnel-no, tunnel-yes, tunnel-unassigned
      Need :  Mandatory
   Default :  tunnel-unassigned

      Name :  encrypted
      Help :  Encrypted applications
      Type :  group
    Groups :  encrypted-yes, encrypted-no, encrypted-unassigned
      Need :  Mandatory
   Default :  encrypted-unassigned
```

The following sample output from the **show ip nbar protocol-attribute** command displays the details of the protocols:

```
Router# show ip nbar protocol-attribute

           Protocol Name :  ftp
                category :  file-sharing
            sub-category :  client-server
       application-group :  ftp-group
                  tunnel :  tunnel-no
               encrypted :  encrypted-no

           Protocol Name :  http
                category :  browsing
            sub-category :  other
       application-group :  other
                  tunnel :  tunnel-no
               encrypted :  encrypted-no

           Protocol Name :  egp
                category :  net-admin
            sub-category :  routing-protocol
       application-group :  other
                  tunnel :  tunnel-no
               encrypted :  encrypted-no

           Protocol Name :  gre
                category :  net-admin
```

```
            sub-category :  tunneling-protocols
       application-group :  other
                  tunnel :  tunnel-yes
               encrypted :  encrypted-no
!
!
!
```

Example: Adding Custom Values for Attributes

The following example shows how to add custom values for the attributes application-group, category, and sub-category:

```
Device> enable
Device# configure terminal
Device(config)# ip nbar attribute application-group custom Home_grown_finance_group "our
finance tools network traffic"
Device(config)# ip nbar attribute category custom dc_backup_category "Data center backup
traffic"
Device(config)# ip nbar attribute sub-category custom hr_sub_category "HR custom applications
 traffic"
Device(config)# exit
```

Examples: Viewing the Information About Custom Values for Attributes

The following sample output from the **show ip nbar attribute-custom** command displays the number of custom values that can be defined, and the custom values that are currently defined for the attributes:

```
Device# show ip nbar attribute-custom

                  Name :  category
                  Help :  category attribute
   Custom Groups Limit :  1
 Custom Groups Created :  dc_backup_category

                  Name :  sub-category
                  Help :  sub-category attribute
   Custom Groups Limit :  1
 Custom Groups Created :  hr_sub_category

                  Name :  application-group
                  Help :  application-group attribute
   Custom Groups Limit :  1
 Custom Groups Created :  Home_grown_finance_group
```

The following sample output from the **show ip nbar attribute category** command displays the details about the Category attribute:

```
Device# show ip nbar attribute category

      Name :  category
      Help :  category attribute
      Type :  group
    Groups :  newsgroup
           :  instant-messaging
           :  net-admin
           :  trojan
```

```
                :  email
                :  file-sharing
                :  industrial-protocols
                :  business-and-productivity-tools
                :  internet-privacy
                :  social-networking
                :  layer3-over-ip
                :  obsolete
                :  streaming
                :  location-based-services
                :  voice-and-video
                :  other
                :  gaming
                :  browsing
                :  dc_backup_category
      Need :  Mandatory
      Default :  other
```

Where to Go Next

Begin configuring NBAR by first enabling Protocol Discovery. To enable Protocol Discovery, see the "Enabling Protocol Discovery" module.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Classifying Network Traffic Using NBAR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Classifying Network Traffic Using NBAR

Feature Name	Releases	Feature Information
Additional PDL Support for NBAR	15.1(2)T	The Additional PDL Support for NBAR feature provides addtional PDLs as part of feature parity between Cisco IOS Release 12.2(18)ZY and Cisco IOS Release 15.1(2)T. The following section includes information about this PDL support extended to Cisco IOS Release 15.1(2)T: NBAR-Supported Protocols.
Distributed Network-based Application Recognition (dNBAR)	12.1(6)E 12.2(4)T 12.2(18)SXF	dNBAR is NBAR used on the Cisco 7500 router with a VIP and on the Catalyst 6500 family of switches with a FlexWAN module or SIP. The implementation of NBAR and dNBAR is identical. The following section provides information about this feature: Information About Classifying Network Traffic Using NBAR.
Enhanced NBAR	15.2(1)T	The Enhanced NBAR feature provides support for additional protocols. The following section includes information about htis feature: NBAR-Supported Protocols.
nBAR: IANA Protocol Extensions Pack1	15.1(3)T	The nBAR: IANA Protocol Extensions Pack1 feature allows NBAR to detect and classify a set of protocols and applications standardized by IANA. The following section provides information about this feature: NBAR-Supported Protocols.
NBAR Categorization and Attributes	15.2(2)T	The NBAR Categorization and Attributes feature provides the mechanism of matching the protocols grouped under specific categories based on the attributes. These categories are available for Class-Based Policy Language (CPL) as a match criteria for application recognition. The following section provides information about this feature: NBAR Categorization and Attributes.

Feature Name	Releases	Feature Information
NBAR—Network-based Application Recognition	12.1(1)E 12.1(5)T 12.2(11)YT 12.2(18)ZY	NBAR is a classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate QoS for that application or traffic with that protocol. The following section provides information about this feature: Information About Classifying Network Traffic Using NBAR. The following commands were introduced or modified: **ip nbar protocol-tagging**, **match protocol citrix**, **match protocol fasttrack**, **match protocol gnutella**, **match protocol http**, **show ip nbar protocol-tagging**.
NBAR2: Add/Rename Static Attributes	15.4(1)T	The custom values enable you to name the attributes based on grouping of protocols. You can create custom values for the attributes application-group, category, and sub-category. The following section provides information about this feature: NBAR Categorization and Attributes. The following commands were introduced or modified: **ip nbar attribute**, **show ip nbar attribute-custom**, and **show ip nbar category**.
NBAR2 GETVPN (Cryptomap) Support	15.4(2)T	GETVPN is supported. The following section provides information about this feature: NBAR Support for GETVPN, on page 15

Glossary

encryption --Encryption is the application of a specific algorithm to data so as to alter the appearance of the data, making it incomprehensible to those who are not authorized to see the information.

dNBAR --Distributed Network-Based Application Recognition. dNBAR is NBAR used on the Cisco 7500 router with a Versatile Interface Processor (VIP) and on the Catalyst 6500 family of switches with a FlexWAN module or serial interface processor (SIP). The implementation of NBAR and dNBAR is identical.

HTTP --Hypertext Transfer Protocol. The protocol used by web browsers and web servers to transfer files, such as text and graphic files.

IANA --Internet Assigned Numbers Authority. An organization operated under the auspices of the Internet Society (ISOC) as a part of the Internet Architecture Board (IAB). IANA delegates authority for IP address-space allocation and domain-name assignment to the InterNIC and other organizations. IANA also maintains a database of assigned protocol identifiers used in the TCP/IP stack, including autonomous system numbers.

LAN --local-area network. A high-speed, low-error data network that covers a relatively small geographic area (up to a few thousand meters). LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited area. LAN standards specify cabling and signaling at the

physical and data link layers of the Open System Interconnection (OSI) model. Ethernet, FDDI, and Token Ring are widely used LAN technologies.

MIME --Multipurpose Internet Mail Extension. The standard for transmitting nontext data (or data that cannot be represented in plain ASCII code) in Internet mail, such as binary, foreign language text (such as Russian or Chinese), audio, and video data. MIME is defined in RFC 2045, *Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies* .

MPLS --Multiprotocol Label Switching. A switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on preestablished IP routing information.

MQC --modular quality of service command-line interface. A CLI that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach policy maps to interfaces. Policy maps are used to apply the appropriate quality of service (QoS) to network traffic.

NBAR --Network-Based Application Recognition. A classification engine that recognizes and classifies a wide variety of protocols and applications. When NBAR recognizes and classifies a protocol or application, the network can be configured to apply the appropriate quality of service (QoS) for that application or traffic with that protocol.

PDLM --Packet Description Language Module. A file that contains Packet Description Language statements used to define the signature of one or more application protocols.

Protocol Discovery --A feature included with NBAR. Protocol Discovery provides a way to discover the application protocols that are operating on an interface.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RTCP --RTP Control Protocol. A protocol that monitors the QoS of an IPv6 Real-Time Transport Protocol (RTP) connection and conveys information about the ongoing session.

RTSP --Real Time Streaming Protocol. A means for enabling the controlled delivery of real-time data, such as audio and video. Sources of data can include both live data feeds, such as live audio and video, and stored content, such as prerecorded events. RTSP is designed to work with established protocols, such as Real-Time Transport Protocol (RTP) and HTTP.

stateful protocol --A protocol that uses TCP and UDP port numbers that are determined at connection time.

static protocol --A protocol that uses well-defined (predetermined) TCP and UDP ports for communication.

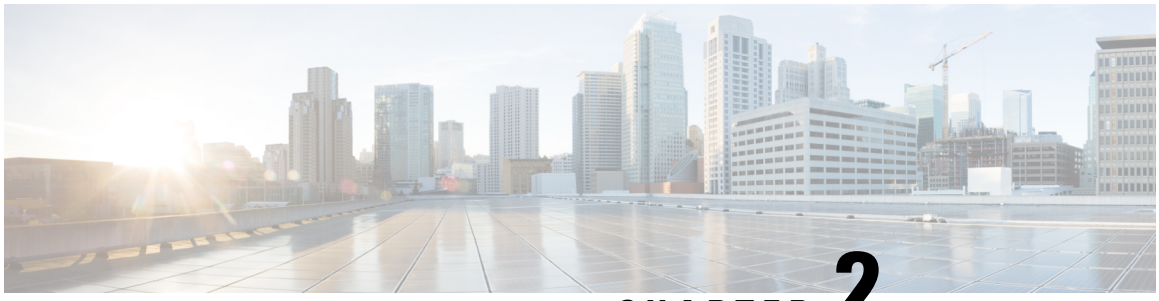
subport classification --The classification of network traffic by information that is contained in the packet payload, that is, information found beyond the TCP or UDP port number.

TCP --Transmission Control Protocol. A connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

tunneling --Tunneling is an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

UDP --User Datagram Protocol. A connectionless transport layer protocol in the TCP /IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768, *User Datagram Protocol* .

WAN --wide-area network. A data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers.



CHAPTER 2

Enabling Protocol Discovery

Network-Based Application Recognition (NBAR) includes a feature called Protocol Discovery. Protocol Discovery provides an easy way to discover the application protocols that are operating on an interface. When you configure NBAR, the first task is to enable Protocol Discovery.

This module contains concepts and tasks for enabling the Protocol Discovery feature.

Prerequisites for Enabling Protocol Discovery

Before enabling Protocol Discovery, read the information in the "Classifying Network Traffic Using NBAR" module.

Information About Protocol Discovery

Protocol Discovery Functionality

NBAR determines which protocols and applications are currently running on your network. NBAR includes a feature called Protocol Discovery. Protocol Discovery provides an easy way of discovering the application protocols that are operating on an interface so that appropriate quality of service (QoS) features can be applied. With Protocol Discovery, you can discover any protocol traffic that is supported by NBAR and obtain statistics that are associated with that protocol.

Protocol Discovery maintains the following per-protocol statistics for enabled interfaces:

- Total number of input packets and bytes
- Total number of output packets and bytes
- Input bit rates

- Output bit rates

The statistics can then be used when you later define classes and traffic policies (sometimes known as policy maps) for each traffic class. The traffic policies (policy maps) are used to apply specific QoS features and functionality to the traffic classes.

How to Configure Protocol Discovery

Enabling Protocol Discovery on an Interface

The **ip nbar protocol-discovery**command is used to enable Protocol Discovery on an interface. With Cisco IOS Release 12.2(18)ZYA, intended for use on the Cisco 6500 series switch that is equipped with a Supervisor 32/PISA, the **ip nbar protocol-discovery** command is supported on both Layer 2 and Layer 3 Etherchannels.

To enable Protocol Discovery on an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip nbar protocol-discovery**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**ip nbar protocol-discovery** **Example:** `Router(config-if)# ip nbar protocol-discovery`	Configures NBAR to discover traffic for all protocols known to NBAR on a particular interface.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Reporting Protocol Discovery Statistics

To display a report of the Protocol Discovery statistics per interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **show ip nbar protocol-discovery** [**interface** *type number*] [**stats** {**byte-count** | **bit-rate** | **packet-count**| **max-bit-rate**}] [**protocol** *protocol-name* | **top-n** *number*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map interface** *type number* **Example:** `Router# show policy-map interface Fastethernet 6/0`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and the interface number.
Step 3	**show ip nbar protocol-discovery** [**interface** *type number*] [**stats** {**byte-count**	**bit-rate**	**packet-count**| **max-bit-rate**}] [**protocol** *protocol-name*	**top-n** *number*] **Example:** `Router# show ip nbar protocol-discovery interface Fastethernet 6/0`	Displays the statistics gathered by the NBAR Protocol Discovery feature. • (Optional) Enter keywords and arguments to fine-tune the statistics displayed.
Step 4	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Enabling Protocol Discovery

Example Enabling Protocol Discovery on an Interface

In the following sample configuration, Protocol Discovery is enabled on Ethernet interface 2/4.

```
Router> enable

Router# configure terminal

Router(config)# interface ethernet 2/4

Router(config-if)# ip nbar protocol-discovery

Router(config-if)# end
```

Example Reporting Protocol Discovery Statistics

The following example displays output from the **show ip nbar protocol-discovery**command for the five most active protocols on an Ethernet interface:

```
Router# show ip nbar protocol-discovery top-n 5

 Ethernet2/0
                           Input                     Output
                           -----                     ------
  Protocol                 Packet Count              Packet Count
                           Byte Count                Byte Count
                           30sec Bit Rate (bps)      30sec Bit Rate (bps)
                           30sec Max Bit Rate (bps)  30sec Max Bit Rate (bps)
--------------------------- ------------------------ ------------------------
   rtp                     3272685                   3272685
                                          242050604                 242050604

                           768000                    768000
                           2002000                   2002000
   gnutella                513574                    513574
                           118779716                 118779716
                           383000                    383000
                           987000                    987000
   ftp                     482183                    482183
                           37606237                  37606237
                           121000                    121000
                           312000                    312000
   http                    144709                    144709
                           32351383                  32351383
                           105000                    105000
                           269000                    269000
   netbios                 96606                     96606
                           10627650                  10627650
                           36000                     36000
                           88000                     88000
   unknown                 1724428                   1724428
```

```
                                534038683               534038683
                                2754000                 2754000
                                4405000                 4405000
Total                           6298724                 6298724
                                989303872               989303872
                                4213000                 4213000
                                8177000                 8177000
```

Where to Go Next

After you enable Protocol Discovery, you have the option to configure NBAR using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). To configure NBAR using the MQC, see the "Configuring NBAR Using the MQC" module.

Additional References

The following sections provide references related to enabling Protocol Discovery.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Configuring NBAR using the MQC	"Configuring NBAR Using the MQC" module
Adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

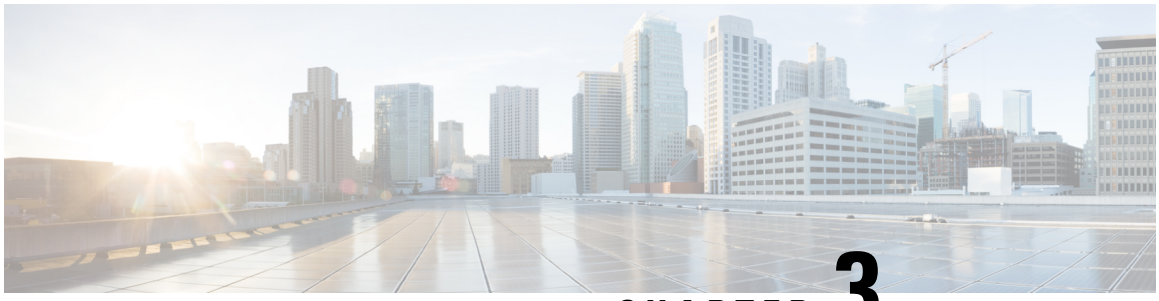
Feature Information for Enabling Protocol Discovery

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Enabling Protocol Discovery

Feature Name	Releases	Feature Information
NBAR--Network-Based Application Recognition	12.2(18)ZYA	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). The following commands were modified: **ip nbar protocol-discovery, show ip nbar protocol-discovery**.



CHAPTER 3

Configuring NBAR Using the MQC

You can configure Network-Based Application Recognition (NBAR) using the functionality of the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). The MQC uses traffic classes and traffic policies (policy maps) to apply QoS features to classes of traffic and applications recognized by NBAR.

This module contains concepts and tasks for configuring NBAR using the MQC.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Configuring NBAR Using the MQC

Before configuring NBAR using the MQC, read the information in the "Classifying Network Traffic Using NBAR" module.

Information About NBAR Coarse-Grain Classification

NBAR and the MQC Functionality

To configure NBAR using the MQC, you must define a traffic class, configure a traffic policy (policy map), and then attach that traffic policy to the appropriate interface. These three tasks can be accomplished by using the MQC. The MQC is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach these traffic policies to interfaces.

In the MQC, the **class-map** command is used to define a traffic class (which is then associated with a traffic policy). The purpose of a traffic class is to classify traffic.

Using the MQC to configure NBAR consists of the following:

- Defining a traffic class with the **class-map** command.
- Creating a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attaching the traffic policy to the interface with the **service-policy** command.

A traffic class contains three major elements: a name, one or more **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands (that is, **match-all** or **match-any**). The traffic class is named in the **class-map** command line; for example, if you enter the **class-map cisco** command while configuring the traffic class in the CLI, the traffic class would be named "cisco."

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the **match** commands. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

Note For NBAR, the **match protocol** commands are used to specify the match criteria, as described in the NBAR and the match protocol Commands, on page 32.

NBAR and the match protocol Commands

NBAR recognizes specific network protocols and network applications that are used in your network. Once a protocol or application is recognized by NBAR, you can use the MQC to group the packets associated with those protocols or applications into classes. These classes are grouped on the basis of whether the packets conform to certain criteria.

For NBAR, the criterion is whether the packet matches a specific protocol or application known to NBAR. Using the MQC, network traffic with one network protocol (citrix, for example) can be placed into one traffic class, while traffic that matches a different network protocol (gnutella, for example) can be placed into another traffic class. Later, the network traffic within each class can be given the appropriate QoS treatment by using a traffic policy (policy map).

You specify the criteria used to classify traffic by using a **match protocol**command. The table below lists some of the available **match protocol**commands and the corresponding protocol or traffic type recognized and supported by NBAR.

Note For a more complete list of the protocol types supported by NBAR, see the "Classifying Network Traffic Using NBAR" module.

Table 3: match protocol Commands and Corresponding Protocol or Traffic Type

match protocol Command[1]	Protocol Type
match protocol (NBAR)	Protocol type supported by NBAR
match protocol citrix	Citrix protocol
match protocol fasttrack	FastTrack peer-to-peer traffic
match protocol gnutella	Gnutella peer-to-peer traffic
match protocol http	Hypertext Transfer Protocol
match protocol rtp	Real-Time Transport Protocol traffic
match protocol unknown [final]	All unknown and/or unclassified traffic

[1] Cisco IOS match protocol commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using.

How to Configure NBAR Using the MQC

Configuring DSCP-Based Layer 3 Custom Applications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *name* **transport** {**tcp** | **udp** | **udp-tcp** }**id** *id*
4. **dscp** *dscp-value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *name* **transport** {**tcp** | **udp** | **udp-tcp** }**id** *id* **Example:** `Device(config)# ip nbar custom mycustom transport tcp id 100`	Specifies the transport protocol to match as TCP, UDP, or both TCP and UDP, and enters custom configuration mode.
Step 4	**dscp** *dscp-value* **Example:** `Device(config-custom)# dscp ef`	Specifies the differentiated service code points (DSCP) value. **Note** In cases where two custom applications have the same filters, the priority is set according to the order of configuration.
Step 5	**exit** **Example:** `Device(config-custom)# exit`	Exits custom configuration mode.

Managing Unclassified and Unknown Traffic

Some protocols require the analysis of more than one packet for NBAR classification. So packets sent until such a classification occurs are considered **unknown**. **unknown final** excludes these temporarily classified packets, and includes only those packets that are determined as unknown after the NBAR classification process.

By default, all traffic not matched to the unknown, are matched to a default class, as is the case with MQC.

Before you begin

Ensure that NBAR is fully configured. If NBAR is configured to match only a partial set of protocols, then all inactivate protocols are considered as unclassified traffic and hence unknown.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] **unknown**
4. **match protocol unknown [final]**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] **unknown** **Example:** `Device(config)# class-map match-all my-unknown`	Creates a class map to be used for matching unknown traffic to a new class and enters class-map configuration mode.
Step 4	**match protocol unknown [final]** **Example:** `Device(config-cmap)# match protocol unknown final`	Configures NBAR to match unknown traffic. • The **unknown** keyword signifies any traffic that is unclassified • The **unknown final** signifies traffic that is determined by NBAR as unknown.
Step 5	**end** **Example:** `Device(config-cmap)# end`	(Optional) Returns to privileged EXEC mode.

You can now configure the following tasks

1. Configuring a Traffic Policy
2. Attaching a Traffic Policy to an Interface or sub-interface

Configuring a Traffic Policy

Traffic that matches a user-specified criterion can be organized into a specific class that can, in turn, receive specific user-defined QoS treatment when that class is included in a policy map.

To configure a traffic policy, perform the following steps.

Note The **bandwidth** command is shown in Step 5. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use. As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA).

Note For Cisco IOS Release 12.2(18)ZY, an existing traffic policy (policy map) cannot be modified if the traffic policy is already attached to the interface. To remove the policy map from the interface, use the **no** form of the **service-policy** command.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map policy1`	Creates or modifies a policy map that can be attached to one or more interfaces and enters policy-map configuration mode. • Enter the name of the policy map.
Step 4	**class** {*class-name*	**class-default**} **Example:** `Device(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change and enters policy-map class configuration mode. • Enter the specific class name or enter the **class-default**keyword.

	Command or Action	Purpose
Step 5	**bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*} **Example:** `Device(config-pmap-c)# bandwidth percent 50` **Example:**	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use. **Note** As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	(Optional) Returns to privileged EXEC mode.

Attaching a Traffic Policy to an Interface or Subinterface

After a policy map is created, the next step is to attach the traffic policy (sometimes called a policy map) to an interface or subinterface. Traffic policies can be attached to either the input or output direction of the interface or subinterface.

Note Depending on the needs of your network, you may need to attach the traffic policy to an ATM PVC, a Frame Relay data-link connection identifier (DLCI), or other type of interface.

To attach a traffic policy (policy map) to an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Device(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**] **Example:** `Device(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	**exit** **Example:** `Device(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Device(config-if)# service-policy input policy1`	Attaches a policy map (traffic policy) to an input or output interface. • Specify either the **input** or **output** keyword, and enter the policy map name.

	Command or Action	Purpose

		Note Policy maps can be configured on ingress or egress Devices. They can also be attached in the input or output direction of an interface. The direction (input or output) and the Device (ingress or egress) to which the policy map should be attached vary according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the Device and the interface direction that are appropriate for your network configuration.
		
		Note After you use the **service-policy** command, you may see two messages similar to the following:
		
		`%PISA-6-NBAR_ENABLED: feature accelerated on input direction of: [interface name and type ]`
		`%PISA-6-NBAR_ENABLED: feature accelerated on output direction of: [interface name and type`
		
		While both of these messages appear, NBAR is enabled in the direction specified by the **input** or **output** keyword only.

Step 7	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying NBAR Using the MCQ

After you create the traffic classes and traffic policies (policy maps), you may want to verify that the end result is the one you intended. That is, you may want to verify whether your traffic is being classified correctly and whether it is receiving the QoS treatment as intended. You may also want to verify that the protocol-to-port mappings are correct.

To verify the NBAR traffic classes, traffic policies, and protocol-to-port mappings, perform the following steps.

SUMMARY STEPS

1. **show class-map** [*class-map-name*]
2. **show policy-map** [*policy-map*]
3. **show policy-map interface** *type number*
4. **show ip nbar port-map** [*protocol-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show class-map** [*class-map-name*] **Example:** `Device# show class-map`	(Optional) Displays all class maps and their matching criteria. • (Optional) Enter the name of a specific class map.
Step 2	**show policy-map** [*policy-map*] **Example:** `Device# show policy-map`	(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. • (Optional) Enter the name of a specific policy map.
Step 3	**show policy-map interface** *type number* **Example:** `Device# show policy-map interface Fastethernet 6/0`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and the interface number.
Step 4	**show ip nbar port-map** [*protocol-name*] **Example:** `Device# show ip nbar port-map`	(Optional) Displays the current protocol-to-port mappings in use by NBAR. • (Optional) Enter a specific protocol name.

Verifying Unknown and Unclassified Traffic Management

To verify the management of unknown and unclassified traffic, perform the following steps.

SUMMARY STEPS

1. **show ip nbar protocol-id unknown**
2. **show ip nbar link-age unknown**
3. **show ip nbar protocol-attribute unknown**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ip nbar protocol-id unknown**
	Example:
	`Device# show ip nbar protocol-id unknown`
	`Protocol Name id type`
	`----------`
	`unknown 1 L7 STANDARD`
		(Optional) Displays protocol classification ID for unknown and unclassified traffic.

Step 2	**show ip nbar link-age unknown** **Example:**	(Optional) Displays the protocol link age for unknown and unclassified traffic.

	Command or Action	Purpose

	`Device# show ip nbar link-age unknown`
	
	`Protocol Link Age (seconds)`
	`unknown 60`	

Step 3	**show ip nbar protocol-attribute unknown**
	Example:
	
	`Device# show ip nbar protocol-attribute unknown`
	
	`Protocol Name : unknown`
	`encrypted : encrypted-no`
	`tunnel : tunnel-no`
	`category : other`
	`sub-category : other`
	`application-group : other`
	`p2p-technology : p2p-tech-no`
		(Optional) Displays list of configured attributes for unknown and unclassified traffic.

Configuration Examples for Configuring DSCP-Based Layer 3 Custom Applications

Example Configuring a Traffic Class

In the following example, a class called cmap1 has been configured. All traffic that matches the citrix protocol will be placed in the cmap1 class.

```
Device> enable

Device# configure terminal

Device(config)# class-map cmap1

Device(config-cmap)# match protocol citrix

Device(config-cmap)# end
```

Example Configuring a Traffic Policy

In the following example, a traffic policy (policy map) called policy1 has been configured. Policy1 contains a class called class1, within which CBWFQ has been enabled.

```
Device> enable

Device# configure terminal
```

```
Device(config)# policy-map policy1

Device(config-pmap)# class class1

Device(config-pmap-c)# bandwidth percent 50

Device(config-pmap-c)# end
```

Note In the above example, the **bandwidth** command is used to enable Class-Based Weighted Fair Queuing (CBWFQ). CBWFQ is only an example of one QoS feature that can be applied in a policy map. Use the appropriate command for the QoS feature that you want to use. As of Cisco IOS Release 12.2(18)ZY, CBWFQ is not supported on the Catalyst 6500 series switch that is equipped with a Supervisor 32/PISA.

Example Attaching a Traffic Policy to an Interface or Subinterface

In the following example, the traffic policy (policy map) called policy1 has been attached to Ethernet interface 2/4 in the input direction of the interface.

```
Device> enable

Device# configure terminal

Device(config)# interface ethernet 2/4

Device(config-if)# service-policy input policy1

Device(config-if)# end
```

Example Verifying the NBAR Protocol-to-Port Mappings

The following is sample output of the **show ip nbar port-map** command. This command displays the current protocol-to-port mappings in use by NBAR. Use the display to verify that these mappings are correct.

```
Device# show ip nbar port-map
port-map bgp      udp 179
port-map bgp      tcp 179
port-map cuseeme  udp 7648 7649
port-map cuseeme  tcp 7648 7649
port-map dhcp     udp 67 68
port-map dhcp     tcp 67 68
```

If the **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the ports assigned to the protocol.

If the **no ip nbar port-map**command has been used, the **show ip nbar port-map** command displays the default ports. To limit the display to a specific protocol, use the *protocol-name* argument of the **show ip nbar port-map** command.

Example: L3 Custom any IP Port

```
Device> enable
Device# configuration terminal
Device (config)# ip nbar custom mycustom transport udp-tcp
Device(config-custom)# dscp ef
Device (config-custom)# exit
```

Where to Go Next

To add application recognition modules (also known as Packet Description Language Modules or PDLMs) to your network, see the "Adding Application Recognition Modules" module.

To classify network traffic on the basis of a custom protocol, see the "Creating a Custom Protocol" module.

Additional References

The following sections provide references related to configuring NBAR using the MQC.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features and functionality on the Catalyst 6500 series switch	"Configuring PFC QoS" chapter of the *Catalyst Supervisor Engine 32 PISA Cisco IOS Software Configuration Guide* , Release 12.2ZY
MQC, traffic policies (policy maps), and traffic classes	"Applying QoS Features Using the MQC" module
CBWFQ	"Configuring Weighted Fair Queueing" module
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Information about enabling Protocol Discovery	"Enabling Protocol Discovery" module
Information about adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring NBAR Using the MQC

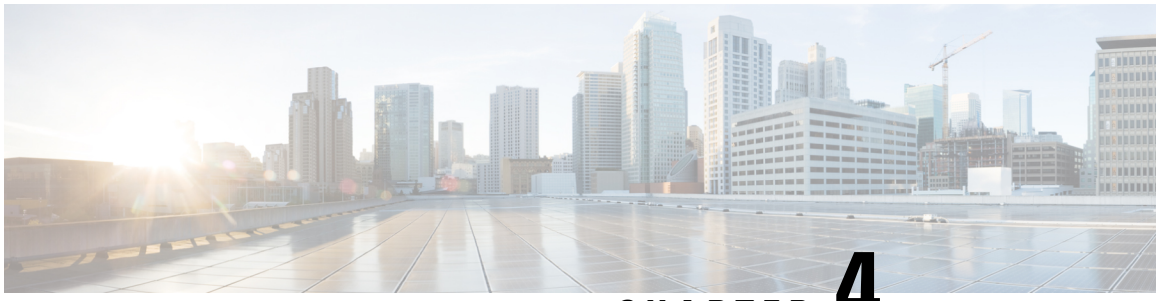
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Configuring NBAR Using the MQC

Feature Name	Releases	Feature Information
NBAR MQC Support for Pre-resolved and Unknown Applications	IOS Release 15.5(1)T IOS XE Release 3.14S	The NBAR MQC Support for Pre-resolved and Unknown Applications feature provides support for matching all unknown and unclassified traffic using MQC. The following commands were modified: **class-map**, **match protocol**
QoS: DirectConnect PDLM	12.4(4)T	Provides support for the DirectConnect protocol and Packet Description Language Module (PDLM). The DirectConnect protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the QoS: DirectConnect PDLM feature:
QoS: Skype Classification	12.4(4)T	Provides support for the Skype protocol. The Skype protocol can now be recognized when using the MQC to classify traffic. **Note** Cisco currently supports Skype Version 1 only. The following sections provide information about the QoS: Skype Classification feature:

Feature Name	Releases	Feature Information
NBAR--BitTorrent PDLM	12.4(2)T	Provides support for the BitTorrent PDLM and protocol. The BitTorrent protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the NBAR-BitTorrent PDLM feature:
NBAR--Citrix ICA Published Applications	12.4(2)T	Enables NBAR to classify traffic on the basis of the Citrix Independent Computing Architecture (ICA) published application name and tag number. The following sections provide information about the NBAR-Citrix ICA Published Applications feature:
NBAR--Multiple Matches Per Port	12.4(2)T	Provides the ability for NBAR to distinguish between values of an attribute within the traffic stream of a particular application on a TCP or UDP port. The following sections provide information about the NBAR-Multiple Matches Per Port feature:
NBAR Extended Inspection for HTTP Traffic	12.3(4)T	Allows NBAR to scan TCP ports that are not well known and identify HTTP traffic that traverses these ports. The following sections provide information about the NBAR Extended Inspection for HTTP Traffic feature:
NBAR Real-Time Transport Protocol Payload Classification	12.2(15)T	Enables stateful identification of real-time audio and video traffic. The following section provides information about the NBAR Real-Time Transport Protocol Payload Classification feature:
NBAR--Network-Based Application Recognition	12.2(18)ZYA	Integrates NBAR and Firewall Service Module (FWSM) functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). Additional protocols are now recognized by NBAR. The following sections provide information about the NBAR feature: The following command was modified: **match protocol (NBAR)**.
NBAR--Network-Based Application Recognition (Hardware Accelerated NBAR)	12.2(18)ZY	Enables NBAR functionality on the Catalyst 6500 series switch that is equipped with a Supervisor 32/programmable intelligent services accelerator (PISA). The following section provides information about the NBAR--Network-Based Application Recognition (Hardware Accelerated NBAR) feature:



CHAPTER 4

MQC Based on Transport Hierarchy

The MQC Based on Transport Hierarchy(TPH) feature enables the use of TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol, for example, an email application over HTTP. A new MQC filter configured within a class-map matches all traffic which has this protocol in the hierarchy.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restrictions for MQC Based on Transport Hierarchy

- The MQC Based on Transport Hierarchy feature is supported only for DNS, HTTP, RTP, and SSL.
- Does not allow adding the match of the protocol and in-app-hierarchy to the same class-map.
- Match protocol http in-app-hierarchy and match protocol rtp in-app-hierarchy are not supported while match protocol attribute tunnel is configured, even on a different class-map.

Information About MQC Based on Transport Hierarchy

MQC Based on Transport Hierarchy Overview

The MQC based on transport hierarchy(TPH) feature enables NBAR to use TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. The TPH of a particular application is the stack of protocols on which the application is delivered. For example, an application is being transported over HTTP and HTTP runs over TCP.

Prior to the configuartion of the MQC based on transport hierarchy(TPH) feature, it is only possible to apply a class-map filter on the final classified protocol using the **match protocol** *protocol-id* class-map filter. However, to apply QoS policies on all the traffic of HTTP, then include all the protocols which run over HTTP into the class-map makes the configuration of such use-cases considerably difficult. A solution for this problem is an in-app-hierarchy class-map filter which uses TPH to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. For example, the rule **match protocol** *http* **in-app-hierarchy** matches if HTTP is present in the hierarchy.

How to Configure MQC Based on Transport Hierarchy

Configuring MQC Based on Transport Hierarchy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match protocol** *protocol-name* **in-app-hierarchy**
5. **end**
6. **configure terminal**
7. **policy-map** *policy-map-name*
8. **class** { *class-name* | **class-default** }
9. **end**
10. **configure terminal**
11. **interface** *type number*
12. **service-policy** { **input** | **output** } *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map match-all C1`	Creates a class map to be used for matching packets to a specified class and enters QoS class-map mode. • Enter the name of the class map.
Step 4	**match protocol** *protocol-name* **in-app-hierarchy** **Example:** `Device(config-cmap)# match protocol http in-app-hierarchy`	Configures the match criterion for a class map on the basis of the specified protocol. The keyword **in-app-hierarchy** matches if the protocol is present in the transport hierarchy. Possible values for *protocol-name*: DNS, HTTP, RTP, SSL
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits class-map mode and returns to privileged EXEC mode.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 7	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map P1`	Specifies the name of the policy map and enters policy-map configuration mode.
Step 8	**class** { *class-name* | **class-default** } **Example:** `Device(config-pmap)# class C1`	Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode.
Step 9	**end** **Example:** `Device(config-cmap)# end`	Exits class-map mode and returns to privileged EXEC mode.
Step 10	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 12	**service-policy** { **input** | **output** } *policy-map-name* **Example:**	Specifies the name of the policy map to be attached to the input or output direction of the interface.

	Command or Action	Purpose
	`Device(config-if)# service-policy input P1`	

Verifying MQC Based on Transport Hierarchy

To verify the MQC Based on Transport Hierarchy feature perform the following steps:

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map interface** *type number* **Example:** `Device# show policy-map interface GigabitEthernet0/0/1`	Displays the packet statistics of all classes that are configured for allservice policies either on the specified interface • Enter the interface type and the interface number.
Step 3	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for MQC Based on Transport Hierarchy

Example: Configuring MQC Based on Transport Hierarchy

The following is an example of the configuring MQC based on Transport Hierarchy feature:

```
Device> enable
Device# configure terminal
Device(config)# class-map match-all C1
Device(config-cmap)# match protocol http in-app-hierarchy
Device(config-cmap)# match protocol youtube
Device(config-cmap)# end
Device# configure terminal
Device(config)# policy-map P1
Device(config-pmap)# class C1
```

```
Device(config-cmap)# end
Device# configure terminal
Device(config)# interface GigabitEthernet 0/0/1
Device(config-if)# service-policy input P1
```

A traffic policy called P1 is configured. P1 contains a class called C1 for which QoS bandwidth limitation is configured as an example. All traffic that has final classification of Youtube with HTTP as a transport will be placed in the C1 class. Other possible transports for Youtube, such as DNS, SSL or RTSP, will not be matched by this class-map

Example: Verifying the MQC Based on Transport Hierarchy configuration

The following is a sample output from the **show policy-map interface** command:

```
Device#  show policy-map interface GigabitEthernet0/0/1

GigabitEthernet0/0/1
  Service-policy input: P1

Class-map: C1 (match-all)
      17 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: protocol http in-app-hierarchy
      Match: protocol youtube

    Class-map: class-default (match-any)
      3 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

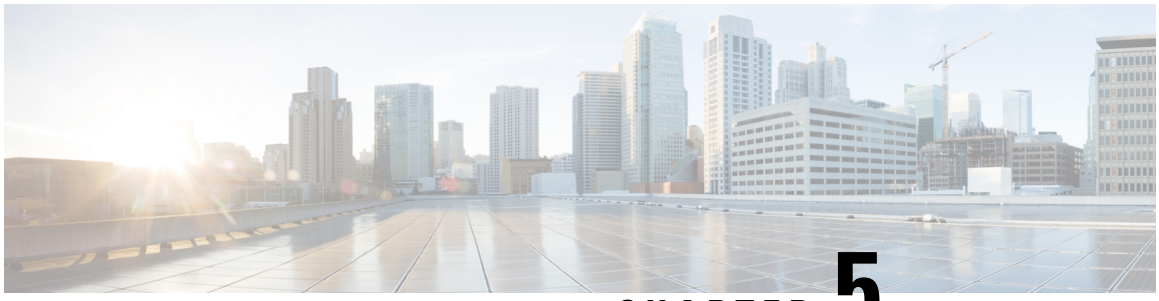
Feature Information for MQC Based on Transport Hierarchy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for MQC Based on Transport Hierarchy

Feature Name	Releases	Feature Information
MQC Based on Transport Hierarchy		The MQC Based on Transport Hierarchy feature enables the use of Transport Hierarchy to apply policies according to a specific underlying protocol, instead of only according to the final classified protocol. A new MQC filter is introduced which can be configured within a class-map. The following command was modified: **match protocol**
Transport Hierarchy support for DNS	Cisco IOS XE Denali 16.3	The **match protocol** CLI can match according to the following protocol types: DNS, HTTP, SSL, and RTP. Example: **match protocol dns in-app-hierarchy**



CHAPTER 5

Adding Application Recognition Modules

Adding application recognition modules (also known as Packet Description Language Modules [PDLMs]) is an optional process. However, adding PDLMs extends the functionality of Network-Based Application Recognition (NBAR) by enabling NBAR to recognize additional protocols on your network.

This module contains concepts and tasks for adding application recognition modules (or PDLMs) to your network.

Prerequisites for Adding Application Recognition Modules

Before adding application recognition modules (or PDLMs), read the information in the "Classifying Network Traffic Using NBAR" module.

Information About Adding Application Recognition Modules

Before adding application recognition modules (or PDLMs), you should understand the following concepts:

PDLM Functionality

A PDLM is a separate file available on Cisco.com. A PDLM is used to add support for a protocol that is currently not available as part of the Cisco IOS software.

A PDLM extends the list of protocols that NBAR can recognize. PDLMs also allow NBAR to recognize new protocols without requiring you to install a new Cisco IOS image or reconfigure your router.

New PDLMs are released by Cisco only and can be loaded from flash memory. Contact your local Cisco representative to request additions or changes to the set of protocols classified by NBAR.

To view a list of currently available PDLMs, go to the following URL:

http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm

PDLM Versioning

A PDLM adds new protocols to the list of protocols that NBAR supports. Before you download a new PDLM, you need to be aware of the following concepts.

Native and Nonnative PDLMs

A native PDLM is a PDLM that is embedded within the Cisco IOS software. You receive it automatically along with the Cisco IOS software.

A nonnative PDLM is not embedded within the Cisco IOS software. You can download it individually from Cisco.com.

Separate Version Numbers

There are separate version numbers associated with the NBAR software and the Cisco IOS software. These version numbers are used together to maintain the PDLM version.

- PDLM version--This is the version of the PDLM (either native or nonnative).
- Cisco IOS NBAR software version--This is the version of NBAR that resides with the Cisco IOS software.

Note Each nonnative PDLM also contains the Cisco IOS NBAR software version in which the PDLM was created.

Internal Module Names

Both the native and nonnative PDLMs contain internal module names. These internal module names are unique and independent. They are used to indicate the protocol that the PDLM module represents (for example, BitTorrent or DirectConnect), and they are used to control the module version number (for example, module version 3).

Required Conditions

In order for a PDLM (either native or nonnative) to be downloaded or overridden, both of the following conditions much be met:

- The module version of the PDLM being downloaded must be higher than the module version currently installed.

For example, if a PDLM called BitTorrent.pdlm (with the internal module name "bittorrent") is currently at version 3, the resident PDLM (either native or nonnative) with the same internal module name ("bittorrent) is overridden as long as the module version is either 1 or 2.

- The Cisco IOS NBAR software version of the PDLM must be less than or equal to the Cisco IOS NBAR software version of the Cisco IOS image.

Note To display the Cisco IOS NBAR software version (of the Cisco IOS image), use the **show ip nbar version** command. For more information about the **show ip nbar version** command, see the Cisco IOS Quality of Service Solutions Command Reference.

How to Add Application Recognition Modules

Downloading a PDLM

A PDLM is used to add support for a protocol that is currently not available as part of the Cisco IOS software. A PDLM extends the functionality of NBAR by enabling NBAR to recognize additional protocols on the network.

To download (install) a PDLM, perform the following steps.

Each PDLM has specific Cisco IOS release requirements and specific restrictions that you need to consider before you download a PDLM. These requirements and restrictions, and other helpful information for installing a particular PDLM, are described in a series of PDLM readme files.

To view a list of currently available PDLMs, or to view the readme files for each PDLM, go to the following URL (Cisco login required):

http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm

Note Use the PDLM readme files in conjunction with the information included in this module.

Before you begin

Protocols introduced when you download a PDLM are commonly added to subsequent Cisco IOS releases. Support for the protocol that you would like to add via a PDLM may already be in your Cisco IOS release. Therefore, before you load a PDLM, review the list of NBAR protocols currently supported by the Cisco IOS release that you are using. To check the list of NBAR protocols supported in your Cisco IOS release, enter the **match protocol ?** command and view the options that appear. The options correspond to the NBAR supported protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar pdlm** *pdlm-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar pdlm** *pdlm-name* **Example:** `Router(config)# ip nbar pdlm flash://citrix.pdlm`	Extends or enhances the list of protocols recognized by NBAR through a Cisco-provided PDLM. • For the *pdlm-name* argument, enter the URL at which the PDLM can be found on the flash card.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Verifying the Downloaded PDLMs

After you download the PDLM, you may want to verify that the PDLM is now on your network. You may also want to check if there are earlier versions of the PDLM already on your network.

To display information about the downloaded PDLMs, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip nbar pdlm**
3. **show ip nbar version** [*pdlm-name*]
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip nbar pdlm** **Example:** `Router# show ip nbar pdlm`	Displays the PDLM in use by NBAR.

	Command or Action	Purpose
Step 3	**show ip nbar version** [*pdlm-name*] **Example:** `Router# show ip nbar version`	Displays information about the version of the NBAR software in your Cisco IOS release or the version of an NBAR PDLM on your Cisco IOS router. • (Optional) Enter the name of the PDLM.
Step 4	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Adding Application Recognition Modules

Example Downloading a PDLM

In the following example. the Citrix PDLM is downloaded to the router from flash memory:

```
Router> enable

Router# configure terminal

Router(config)# ip nbar pdlm flash://citrix.pdlm

Router(config)# end
```

Example Verifying the Downloaded PDLMs

You can use the output of the **show ip nbar pdlm** command and the **show ip nbar version**command to verify information about the downloaded PDLMs on your network.

Sample show ip nbar pdlm Command Output

In this example of the **show ip nbar pdlm** command, the Citrix PDLM has been loaded from flash memory:

```
Router# show ip nbar pdlm

The following PDLMs have been loaded:
flash://citrix.pdlm
```

Sample show ip nbar version Command Output

The following is sample output from the **show ip nbar version**command:

```
Router# show ip nbar version
NBAR software version:  3
1   base                 Mv: 2
2   ftp                  Mv: 2
3   http                 Mv: 7, Nv: 3; slot1:http_vers.pdlm
4   static-port          Mv: 6
5   tftp                 Mv: 1
6   exchange             Mv: 1
7   vdolive              Mv: 1
8   sqlnet               Mv: 1
9   rcmd                 Mv: 1
10  netshow              Mv: 1
11  sunrpc               Mv: 2
12  streamwork           Mv: 1
13  citrix               Mv: 5
14  napster              Mv: 2
15  fasttrack            Mv: 2
16  gnutella             Mv: 1
17  kazaa                Mv: 6, Nv: 3; slot1:kazaa2_vers.pdlm
18  custom-protocols     Mv: 1
19  rtsp                 Mv: 1
20  rtp                  Mv: 2
21  mgcp                 Mv: 1
22  skinny               Mv: 1
23  h323                 Mv: 1
24  sip                  Mv: 1
25  rtcp                 Mv: 1
```

The table below describes the fields shown in the display.

Table 6: show ip nbar version Field Descriptions

Field	Description
NBAR software version	NBAR software version that is running in the current Cisco IOS software. In this particular example, version 3 is shown.
Mv	Resident Module Version. The Resident Module Version is the version of the NBAR-supported PDLM protocol and, therefore, varies by protocol. The Resident Module Version of TFTP, for example, is 1.
Nv	Minimum version of the NBAR software that is required to load a nonnative PDLM. This number is available only for nonnative PDLMs that were loaded onto the router, such as the Kazaa PDLM (protocol 17); in that case, the Nv version is 3.

Where to Go Next

To classify network traffic on the basis of a custom protocol, see the "Creating a Custom Protocol" module.

Additional References

The following sections provide references related to adding application recognition modules.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Information about enabling Protocol Discovery	"Enabling Protocol Discovery" module
Configuring NBAR using the MQC	"Configuring NBAR Using the MQC" module
Creating a custom protocol	"Creating a Custom Protocol" module

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

FeatureInformationforAddingApplicationRecognitionModules

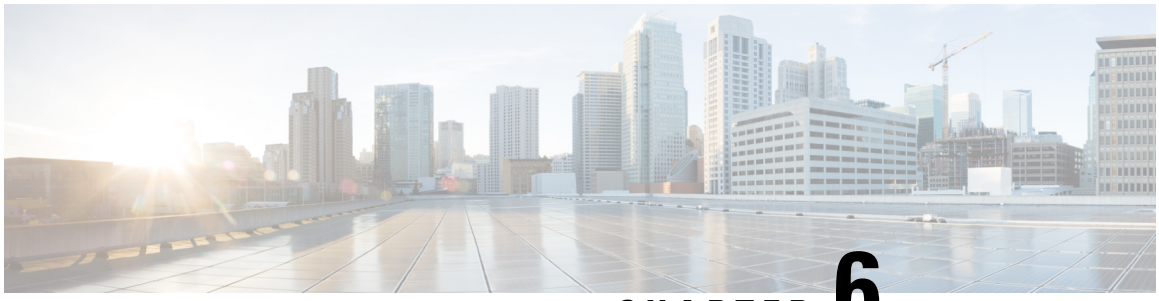
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for Adding Application Recognition Modules

Feature Name	Releases	Feature Information
QoS: DirectConnect PDLM	12.4(4)T	Provides support for the DirectConnect protocol and PDLM. The DirectConnect protocol can now be recognized when using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) to classify traffic. The following sections provide information about the QoS: DirectConnect PDLM feature:

Feature Name	Releases	Feature Information
NBAR - BitTorrent PDLM	12.4(2)T	Provides support for the BitTorrent PDLM and protocol. The BitTorrent protocol can now be recognized when using the MQC to classify traffic. The following sections provide information about the NBAR - BitTorrent PDLM feature:
NBAR PDLM Versioning	12.3(4)T	Enables the ability to verify the Cisco IOS and NBAR PDLM versions for ensuring software compatibility. The following sections provide information about the NBAR PDLM Versioning feature: • Information About Adding Application Recognition Modules • How to Add Application Recognition Modules.



CHAPTER 6

NBAR Protocol Pack

The NBAR Protocol Pack feature provides an easy method to load a protocol pack, which is a single compressed file that contains multiple Protocol Description Language (PDL) files and a manifest file. Before this feature was introduced, PDLs had to be loaded separately. With the Network-Based Application Recognition (NBAR) protocol pack, a set of required protocols can be loaded on the device, helping NBAR recognize additional protocols for classification on your network.

The protocol pack manifest file contains a description of the protocol pack. Protocol Description Language Modules (PDLMs) are used to add support for a protocol that is currently not available as part of the Cisco software.

Prerequisites for the NBAR Protocol Pack

The protocol pack must be copied to your local disk to avoid any errors after rebooting.

Note It is strongly recommended to load the NBAR protocol pack that is the exact match for the NBAR engine, and also load the latest rebuild of Cisco software.

Restrictions for the NBAR Protocol Pack

Only one protocol pack is supported per device.

Information About the NBAR Protocol Pack

NBAR Protocol Pack Overview

NBAR protocol packs are software packages that update the NBAR protocol support on a device without replacing the Cisco software on the device. An NBAR protocol pack contains a set of signatures that is supported by NBAR.

Protocol packs have the following characteristics:

- They are easy to load.
- They are easy to upgrade to a higher version protocol pack or revert to a lower version protocol pack.
- They provide only the required set of protocols.

Cisco provides users with two different protocol packs—the Standard Protocol Pack and the Advanced Protocol Pack—depending on whether they are using an unlicensed or licensed Cisco image.

Cisco provides a specific identity number for the organization (also known as the "publisher") that creates the protocol packs and uses Cisco tools and processes to create new protocol packs. The organization that creates the protocol pack owns the pack.

Cisco provides the Advanced Protocol Pack as the base protocol pack with a licensed Cisco image on a device. The Advanced Protocol Pack has the complete set of Protocol Description Language (PDL) files available for a release. On the Advanced Protocol Pack, only a PDLM with the NAME field as Advanced Protocol Pack can be loaded.

Cisco provides the Standard Protocol Pack as the base protocol pack with an unlicensed Cisco image on a device. The Standard Protocol Pack has limited features and functionality. Some of the features, such as Category and Attributes, Field Extraction, and Tunneled Classification, are not supported. On the Standard Protocol Pack, only a PDLM with the NAME field as Standard Protocol Pack can be loaded.

To view the list of protocols supported in a protocol pack, see NBAR Protocol Library.

The NBAR taxonomy file contains the information such as common name, description, underlying protocol, for every protocol that is available in the protocol pack. Use the **show ip nbar protocol-pack active taxonomy**, **show ip nbar protocol-pack inactive taxonomy** , and **show ip nbar protocol-pack loaded taxonomy** commands to view the taxonomy file for an active, inactive, and all loaded protocol-packs respectively.

The nbar taxonomy file generally contains the information for more than 1000 protocols, and the taxonomy file size is ~2 MB. It is recommended to redirect the output from the **show ip nbar protocol-pack** [**active** | **inactive** | **loaded**] taxonomy command to a file by using the redirect output modifier, for example, **show ip nbar protocol-pack active taxonomy** | **redirect harddisk:***nbar_taxonomy.xml*.

How to Load the NBAR Protocol Pack

Loading the NBAR2 Protocol Pack

Before you begin

Loading a new Protocol Pack requires an advanced license.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar protocol-pack** *protocol-pack* [**force**]
4. **exit**
5. **show ip nbar protocol-pack** {protocol-pack | **active**} [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar protocol-pack** *protocol-pack* [**force**] **Example:** `Device(config)# ip nbar protocol-pack harddisk:defProtoPack`	Loads the protocol pack. • Use the **force** keyword to specify and load a Protocol Pack of a lower version, which is different from the base protocol pack version. Doing so also removes any configurations that are not supported by the lower version Protocol Pack.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show ip nbar protocol-pack** {protocol-pack | **active**} [**detail**] **Example:**	Displays the protocol pack information. • Verify the loaded protocol pack version, publisher, and other details using this command.

	Command or Action	Purpose
	`Device(config)# show ip nbar protocol-pack active`	• Use the *protocol-pack* argument to display information about the specified protocol pack. • Use the **active** keyword to display active protocol pack information. • Use the **detail** keyword to display detailed protocol pack information.

Configuration Examples for the NBAR2 Protocol Pack

Examples: Loading the NBAR Protocol Pack

The following example shows how to load an NBAR protocol pack named defProtoPack from the hard disk:

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack harddisk:defProtoPack
Device(config)# exit
```

The following example shows how to revert to the base image version of the NBAR protocol pack:

```
Device> enable
Device# configure terminal
Device(config)# default ip nbar protocol-pack
Device(config)# exit
```

The following example shows how to use the **force** keyword to load a protocol pack of a lower version:

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack harddisk:olddefProtoPack force
Device(config)# exit
```

Examples: Verifying the Loaded NBAR Protocol Pack

The following sample output from the **show ip nbar protocol-pack** command shows information about the active protocol pack with a licensed Cisco image on a device:

```
Device# show ip nbar protocol-pack active

ACTIVE protocol pack:
Name:                           Advanced Protocol Pack
Version:                        1.0
Publisher:                      Cisco Systems Inc.
```

The following sample output from the **show ip nbar protocol-pack** command shows detailed information about the active protocol pack with a licensed Cisco image on a device:

```
Device# show ip nbar protocol-pack active detail
```

```
ACTIVE protocol pack:
Name:                            Advanced Protocol Pack
Version:                         1.0
Publisher:                       Cisco Systems Inc.
Protocols:
base                             Mv: 4
ftp                              Mv: 5
http                             Mv: 18
static                           Mv: 6
socks                            Mv: 2
nntp                             Mv: 2
tftp                             Mv: 2
exchange                         Mv: 3
vdolive                          Mv: 1
sqlnet                           Mv: 2
netshow                          Mv: 3
sunrpc                           Mv: 3
streamwork                       Mv: 2
citrix                           Mv: 11
fasttrack                        Mv: 3
gnutella                         Mv: 7
kazaa2                           Mv: 11
```

The following sample output from the **show ip nbar protocol-pack** command shows the protocol pack information of a licensed Cisco image present at the specified device location:

```
Device# show ip nbar protocol-pack disk:0ppsmall_higherversion

Name:                            Advanced Protocol Pack
Version:                         2.0
Publisher:                       Cisco Systems Inc.
```

The following sample output from the **show ip nbar protocol-pack** command shows detailed protocol pack information of a licensed Cisco image present at the specified disk location on a device:

```
Device# show ip nbar protocol-pack disk:0ppsmall_higherversion detail

Name:                            Advanced Protocol Pack
Version:                         2.0
Publisher:                       Cisco Systems Inc.
Protocol Pack contents:
iana                        Mv: 1
base                        Mv: 4
tftp                        Mv: 2
```

The following sample output from the **show ip nbar protocol-pack** command shows information about the active protocol pack with an unlicensed Cisco image on a device:

```
Device# show ip nbar protocol-pack active

ACTIVE protocol pack:
Name:                            Standard Protocol Pack
Version:                         1.0
Publisher:                       Cisco Systems Inc.
```

Example: Viewing the NBAR2 Taxonomy Information

The following sample output from the **show ip nbar protocol-pack active taxonomy** command shows the information about the protocols in the active Protocol Pack:

```
Device# show ip nbar protocol-pack active taxonomy

Protocol Pack Taxonomy for Advanced Protocol Pack:
<?xml version="1.0"?>
<NBAR2-Taxonomy>
  <protocol>
    <name>active-directory</name>
    <engine-id>7</engine-id>
    <enabled>true</enabled>
    <selector-id>473</selector-id>
    <help-string>Active Directory Traffic</help-string>
    <global-id>L7:473</global-id>
    <common-name>Active Directory</common-name>
    <static>false</static>
    <attributes>
      <category>net-admin</category>
      <application-group>other</application-group>
      <p2p-technology>false</p2p-technology>
      <tunnel>false</tunnel>
      <encrypted>false</encrypted>
      <sub-category>network-management</sub-category>
    </attributes>
    <ip-version>
      <ipv4>true</ipv4>
      <ipv6>true</ipv6>
    </ip-version>

<references>http://www.microsoft.com/en-us/server-cloud/windows-server/active-directory.aspx</references>

    <id>1194</id>
    <underlying-protocols>cifs,ldap,ssl,ms-rpc</underlying-protocols>
    <long-description-is-final>true</long-description-is-final>
    <long-description>a directory service created by Microsoft for Windows domain networks,
 responsible for authenticating and authorizing all users and computers within a network
of Windows domain type, assigning and enforcing security policies for all computers in a
network and installing or updating software on network computers</long-description>
    <pdl-version>1</pdl-version>
    <uses-bundling>false</uses-bundling>
  </protocol>
  <protocol>
    <name>activesync</name>
    <engine-id>7</engine-id>
    <enabled>true</enabled>
    <selector-id>490</selector-id>
    <help-string>Microsoft Activesync protocol </help-string>
    <global-id>L7:490</global-id>
    <common-name>ActiveSync</common-name>
    <static>false</static>
    <attributes>
      <category>business-and-productivity-tools</category>
      <application-group>other</application-group>
      <p2p-technology>false</p2p-technology>
      <tunnel>false</tunnel>
      <encrypted>true</encrypted>
      <sub-category>client-server</sub-category>
    </attributes>
    <ip-version>
      <ipv4>true</ipv4>
      <ipv6>true</ipv6>
    </ip-version>
   <references>http://msdn.microsoft.com/en-us/library/dd299446(v=exchg.80).aspx</references>

    <id>1419</id>
```

```
    <underlying-protocols>http</underlying-protocols>
    <long-description-is-final>true</long-description-is-final>
    <long-description>ActiveSync is a mobile data synchronization technology and protocol
based on HTTP, developed by Microsoft. There are two implementations of the technology: one
 which synchronizes data and information with handheld devices with a specific desktop
computer, and another technology, commonly known as Exchange ActiveSync (or EAS), which
provides push synchronization of contacts, calendars, tasks, and email between
ActiveSync-enabled servers and devices.</long-description>
    <pdl-version>1</pdl-version>
    <uses-bundling>false</uses-bundling>
  </protocol>
  .
  .
  .
  .
```

Additional References for NBAR2 Protocol Pack

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS LAN Switching commands	Cisco IOS LAN Switching Command Reference
Cisco IOS QoS configuration information	QoS Configuration Guide

Standards and RFCs

Standards/RFCs	Document Title
RFC 3551	RTP Profile for Audio and Video Conferences with Minimal Control
RFC 6101	The Secure Sockets Layer (SSL) Protocol Version 3.0

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

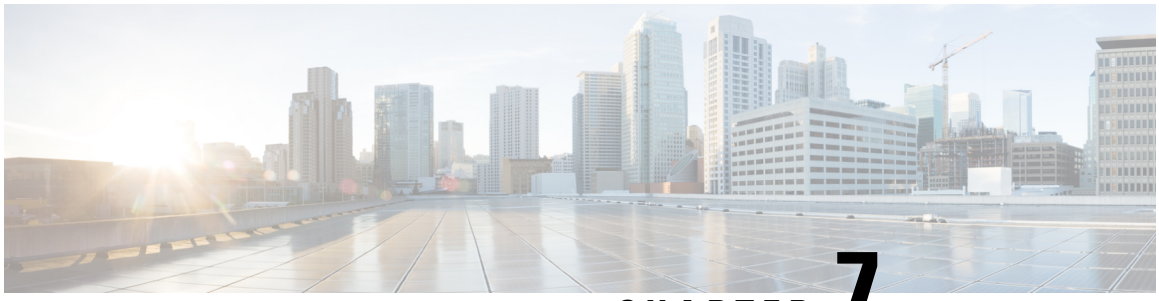
Feature Information for the NBAR Protocol Pack

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for the NBAR Protocol Pack

Feature Name	Releases	Feature Information
NBAR Protocol Pack	15.2(2)T	The NBAR Protocol Pack feature provides an easy method to configure the protocol pack, which is a set of protocols developed and packed together. The following commands were introduced or modified: **default ip nbar protocol-pack**, **ip nbar protocol-pack**, and **show ip nbar protocol-pack**.
NBAR2: Integrate NBAR Taxonomy into the Router	15.4(1)T	The NBAR taxonomy contains the information such as common name, description, underlying protocol, for every protocol that is available in the protocol pack. The following section provides information about this feature: NBAR Protocol Pack Overview, on page 62. The following commands were introduced or modified: **show ip nbar protocol-pack**.



CHAPTER 7

NBAR Protocol Pack Auto Update

Cisco provides periodic updates of NBAR2 Protocol Packs for Cisco IOS releases designated as long-lived, to improve NBAR2 traffic recognition capabilities on an ongoing basis. The Protocol Pack Auto Update feature helps to automate the process of updating any number of participating routers with the latest compatible Protocol Pack.

Overview

Protocol Pack Auto Update streamlines Protocol Pack administrative tasks. It enables network administrators to reduce the repetitive tasks in updating Protocol Packs across a large number of routers in a network.

Rather than operating on each router individually, administrators provide Protocol Pack updates through a centralized "Auto Update" server that stores downloaded Protocol Pack installation files for use by the various routers in the network, and controls the scheduling of updates. The process is controlled through a single configuration file on the server.

After the feature is set up, routers in the network that have Auto Update enabled check the server periodically. If a more up-to-date, compatible Protocol Pack is available, the router downloads the Protocol Pack file and installs it automatically.

Protocol Pack Auto Update – Major Topics

Topic	Section
Deployment	NBAR Protocol Pack Auto Update Deployment, on page 70
Maintenance	Keeping Protocol Packs Up-to-Date, on page 76
Router Procedures	Enabling Protocol Pack Auto Update, on page 77 Disabling Protocol Pack Auto Update, on page 78 Initiating Immediate Protocol Pack Update, on page 78 Displaying Protocol Pack Auto Update Information, on page 79

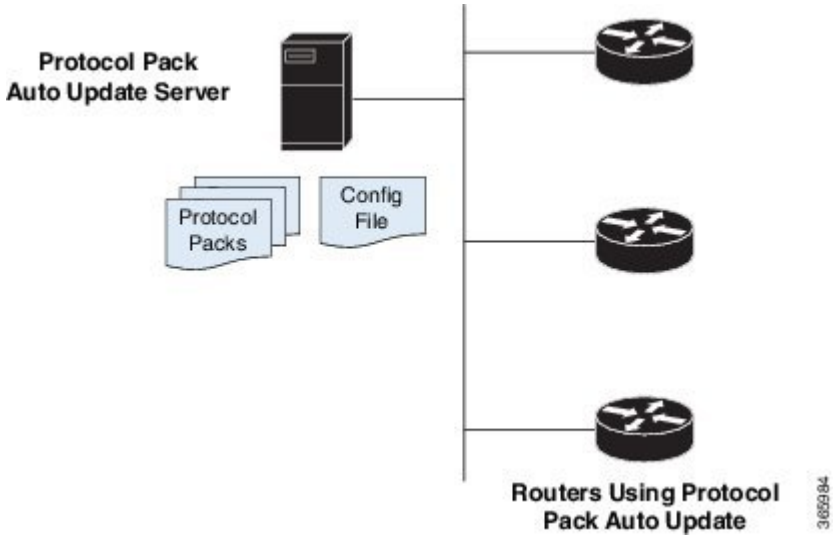
• Configuring Local Protocol Pack Auto Update Settings on a Router, on page 80

NBAR Protocol Pack Auto Update Deployment

Deployment Overview

To deploy Protocol Pack Auto Update in a network, set up an Auto Update server, download the Protocol Pack files for your routers, and create a configuration file customized to your needs. Then simply enable Auto Update on any number of routers within your network as described below.

Figure 1: Protocol Pack Auto Update – Server and Participating Routers



Elements of Protocol Pack Auto Update

- **Protocol Pack Auto Update server**:
  - Downloaded Protocol Pack installation files for routers using Auto Update
  - Configuration file (NBAR_PROTOCOL_PACK_DETAILS.json)
  - Protocol Pack Auto Update log files
- **Routers**: One or more routers with Protocol Pack Auto Update enabled.

  See Enabling Protocol Pack Auto Update, on page 77.

Deployment Steps

1. Set up a Protocol Pack Auto Update server in a location reachable by all routers using Auto Update. (Some CLI commands and output refer to this as the "source-server.")

   See Setting Up a Server for Protocol Pack Auto Update, on page 71.
2. On participating routers, enable Protocol Pack Auto Update.

   See Enabling Protocol Pack Auto Update, on page 77.

Example:

```
Device#configure terminal
Device(config)#ip nbar protocol-pack-auto-update
Device(config-pp-auto-update)#source-server tftp://10.20.300.400/NbarAutoUpdate
Device(config-pp-auto-update)#exit
```

3. (Optional) By default, each router using Auto Update uses the settings provided in the configuration file on the Auto Update server. If required, use Protocol Pack Auto Update CLI commands on an individual router to override the default settings.

   See Configuring Local Protocol Pack Auto Update Settings on a Router, on page 80.

Setting Up a Server for Protocol Pack Auto Update

The Protocol Pack Auto Update server contains the configuration file that controls the feature functionality, and stores the Protocol Pack installation files. To set up the server, use the following procedure.

1. Set up a server in a network location reachable by all participating routers. Make note of the server IP address, to include it in the configuration file.
2. On the server, create the parent directory for storing the configuration file and Protocol Pack installation files.

   ```
   /NbarAutoUpdate/pp_server/
   ```

3. Within the parent directory, /NbarAutoUpdate/pp_server/, create the subdirectories for storing Protocol Pack installation files, organized by platform type.

   ```
   /NbarAutoUpdate/pp_server/asr
   /NbarAutoUpdate/pp_server/csr
   /NbarAutoUpdate/pp_server/isr
   /NbarAutoUpdate/pp_server/isr4k
   /NbarAutoUpdate/pp_server/other
   ```

4. Download the latest Protocol Pack installation files that will be required for the routers using Auto Update. See NBAR2 Protocol Pack Library for information about Protocol Packs, including supported platforms. Download the files using the Download Software tool.
5. Store the Protocol Pack files on the server, in subdirectories of /NbarAutoUpdate/pp_server/.
   - **ASR** directory – Protocol Pack files for Cisco ASR Series devices.
   - **CSR** directory – Protocol Pack files for Cisco CSR Cloud Services Routers.
   - **ISR** directory – Protocol Pack files for Cisco ISR Generation 2 (ISRG2) devices operating with Cisco IOS 15.x releases (not IOS XE).
   - **ISR4K** directory – Protocol Pack files for Cisco ISR4000 Series routers.
   - **OTHER** directory – Protocol Pack files for devices not included in more specific categories.
6. Create the Auto Update JSON-format configuration file, as described in Protocol Pack Auto Update Configuration File, on page 72 and store the file in the Auto Update parent directory:

   ```
   /NbarAutoUpdate/pp_server/NBAR_PROTOCOL_PACK_DETAILS.json
   ```

Multiple Servers Option

It is strongly recommended to use a single server for the Auto Update configuration file and Protocol Pack installation files. However, it is possible to store the Protocol Pack files on a separate server. If doing this, specify the separate server location in the configuration file, where the path to Protocol Pack files is configured.

Protocol Pack Auto Update Configuration File

The Protocol Pack Auto Update configuration file is a JSON-format file, with the required filename NBAR_PROTOCOL_PACK_DETAILS.json. It is stored on the Protocol Pack Auto Update server in the Auto Update parent directory:

```
/NbarAutoUpdate/pp_server/NBAR_PROTOCOL_PACK_DETAILS.json
```

The configuration file specifies:

- Server address
- Locations of the downloaded Protocol Pack files
- NBAR software version for each Protocol Pack file
- Schedule for routers using Auto Update to check the server for updates

Protocol Pack File Locations

The configuration file provides the path for each downloaded Protocol Pack file stored on the server. Routers using Auto Update download the Protocol Pack files from these locations and install them automatically.

The location of each Protocol Pack file is specified by combining the server address, base directory, and specific file path.

- The "protocol-pack-server" section of the configuration file provides the address and base directory.
- The "nbar_pp_files" section provides the paths to individual Protocol Pack installation files.

For example, if the address and base directory are:

```
tftp://10.20.200.1/NbarAutoUpdate/pp_server/
```

...and the Protocol Pack file location is:

```
asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack
```

...then the complete path to the file is:

```
tftp://10.20.200.1/NbarAutoUpdate/pp_server/asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack
```

A router using Auto Update would use this complete path to download the file from the server.

Organization of the Protocol Pack Locations

The "nbar_pp_files" section of the configuration file lists the Protocol Pack files available on the server. Subsections correspond to the directories in which Protocol Packs are stored on the Protocol Pack Auto Update server. Typical subsections include.

- **ASR** – Protocol Pack files for Cisco ASR Series devices.
- **CSR** – Protocol Pack files for Cisco CSR Cloud Services Routers.
- **ISR** – Protocol Pack files for Cisco ISR Generation 2 (ISRG2) devices operating with Cisco IOS 15.x releases (not IOS XE).
- **ISR4K** – Protocol Pack files for Cisco ISR4000 Series routers.
- **OTHER** – Protocol Pack files for devices not included in more specific categories.

Example of the nbar_pp_files section of a configuration file:

```
"nbar_pp_files": {
    "ASR": {
      "23":"asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack"
    },
    "ISR": {
      "23":"isr/pp-adv-isrg2-155-3.M2-23-19.1.0.pack"
    },
    "ISR4K": {
      "23":"pp-adv-isr4000-155-3.Sa4-23-32.1.0.pack",
      "27":"pp-adv-isr4000-163.2-27-35.0.0.pack",
      "31":"pp-adv-isr4000-166.2-31-35.0.0.pack"
    },
    "OTHER": {
      "23":"other/pp-adv-isr4000-155-3.Sa4-23-32.1.0.pack"
    }
}
```

NBAR Software Version Specified for Each Protocol Pack File

Each Protocol Pack installation file is compatible with a specific NBAR software version. The version number typically appears in the filename of the Protocol Pack installation file. For example, the following Protocol Pack 20.0.0 installation file works with NBAR version 23:

```
pp-adv-asr1k-155-3.S2-23-20.0.0.pack
```

In the configuration file, each line that specifies a Protocol Pack installation file location also indicates the matching NBAR software version. When adding Protocol Pack installation file locations, be sure to specifiy the correct NBAR software version for the file. Example:

```
"23":"asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack"
```

Tip Use the **show ip nbar version** command on a router to display the current NBAR software version of the installed OS.

```
Device#show ip nbar version
NBAR software version: 23
NBAR minimum backward compatible version: 21
...
```

Same Router Type, Different Versions of NBAR2

Identical routers running different OS versions may have different versions of NBAR2 and therefore require different Protocol Pack versions—for example, two Cisco ISR 4451 routers, one operating with Cisco IOS XE 3.13 and the other with 3.16. Download the correct Protocol Pack files for both and store them on the Auto Update server.

Configuration File Parameters

The following configuration file parameters provide the default Protocol Pack Auto Update behavior. Individual routers using Auto Update may override these parameters using local CLI commands.

Parameter	Description
protocol-pack-server	(Mandatory) Location of protocol pack server. Example: tftp://10.20.200.1/NbarAutoUpdate/pp_server/
nbar_pp_files	(Mandatory) Provides file locations for protocol pack files for various platforms and NBAR versions, identified by NBAR software version number.
schedule {**daily** | **weekly**: | **monthly**:} [*day*] {**hh**: *hh*, **mm**: *mm*}	Schedule for the Auto Update upgrade interval. Routers using Auto Update check regularly for updates at the scheduled time. • monthly: Day of the month • weekly: Day of the week (0 to 6) • hh: Hour (24-hour time) • mm: Minute The actual run time depends on the **update-window** option. Default: Daily at 00:00
update-window	Maintenance window (in minutes) for NBAR protocol pack auto-update to operate within. The maintenance window is scheduled according to the time configured by the **schedule** parameters. Default: 60
clear-previous	**true**: Causes unneeded Protocol Pack files to be removed after a cool-down period. **false**: Configures the feature to not remove any files. Default: enable

Parameter	Description
force-upgrade	**true**: New Protocol Pack updates will be applied with the **force** flag. **false**: New Protocol Pack updates will not be applied with the **force** flag. Default: disable

Configuration File: Minimal Example

This example of a minimal configuration file contains only the top-level nbar_auto_update_config section, and mandatory fields.

Because no schedule is configured, routers use the default schedule of checking daily at 00:00. The example specifies one Protocol Pack file for each of four platform types.

```
{
"nbar_auto_update_config":{
  "protocol-pack-server":"tftp://10.20.200.1/NbarAutoUpdate/pp_server/"
  },
  "nbar_pp_files":{
    "ASR":{"23":"asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack"},
    "CSR":{"23":"csr/pp-adv-csr1000v-155-3.S2-23-21.0.0.pack"},
    "ISR":{"23":"isr/pp-adv-isrg2-155-3.M2-23-19.1.0.pack"},
    "ISR4K":{"31":"pp-adv-isr4000-166.2-31-35.0.0.pack"}
  }
}
```

Configuration Files: Typical Example

This example of a typical configuration file contains the top-level nbar_auto_update_config section, plus mandatory and optional fields.

- The Protocol Pack Auto Update server address is 10.20.200.1.
- The **schedule** section specifies the update schedule as weekly on Saturdays at 2:30 AM. Routers using Auto Update check at this scheduled time for any available updates.

  Saturday is indicated by the **weekly** value of **6**. The numbering system for days of the week is 0-6, where 0=Sunday and 6=Saturday.

  hh and **mm** specify an update time of 2:30 AM .
- In the **nbar_pp_files** section, the NBAR version number (for example, 23) at the beginning of a line must match the NBAR version number that appears in the Protocol Pack filename.

```
{
  "nbar_auto_update_config": {
    "protocol-pack-server": "tftp://10.20.200.1/NbarAutoUpdate/pp_server/",
    "update-window":0,
    "force-upgrade":true,
    "clear-previous":true,
    "schedule": {
      "weekly": 6,
      "hh": 02,
      "mm": 30
    },
```

```
  },
  "nbar_pp_files": {
    "ASR": {
      "23":"asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack",
    },
    "CSR": {
      "23":"csr/pp-adv-csr1000v-155-3.S2-23-21.0.0.pack"
     },
    "ISR": {
      "23":"isr/pp-adv-isrg2-155-3.M2-23-18.0.0.pack",
      "23":"isr/pp-adv-isrg2-155-3.M2-23-19.1.0.pack"
    },
    "ISR4K":{
      "31":"pp-adv-isr4000-166.2-31-35.0.0.pack"
    }
  }
}
```

Keeping Protocol Packs Up-to-Date

New Protocol Pack Releases

When new Protocol Pack releases become available:

1. Download the new Protocol Pack installation files for the router models in the network using Auto Update.
2. Store the Protocol Pack files in the correct directories on the server.
3. Update the configuration file to include the new Protocol Pack files.

When Upgrading a Router OS

Protocol Pack installation files typically are compatible with a specific platform type running a specific Cisco IOS release.

After upgrading the OS of a router that is using Protocol Pack Auto Update:

1. Use the **show ip nbar version** command to display the NBAR software version. In the following example, the NBAR software version is 23.

```
Device#show ip nbar version

NBAR software version:  23
NBAR minimum backward compatible version:  21

Loaded Protocol Pack(s):

Name:                           Advanced Protocol Pack
Version:                        14.0
Publisher:                      Cisco Systems Inc.
NBAR Engine Version:            23
State:                          Active
```

2. If the NBAR software version has changed, check whether a more up-to-date compatible Protocol Pack is available for the release. (See the NBAR2 Protocol Library page for information about Protocol Pack release compatibility.)
3. If so, download the new Protocol Pack installation file to provide to routers using Auto Update.

4. Store the Protocol Pack file in the correct directory on the server.
5. Update the configuration file to include the new Protocol Pack file.

   Ensure that the new line in the configuration file is in the correct location, and that the specified NBAR2 version number matches the version number in the Protocol Pack filename.

   ```
   "23":"asr/pp-adv-asr1k-155-3.S2-23-20.0.0.pack"
   ```

Enabling Protocol Pack Auto Update

Enabling Protocol Pack Auto Update on a router requires:

- Enabling the feature
- Specifying the Protocol Pack Auto Update server to use, or ensuring that it has been specified already

SUMMARY STEPS

1. **configure terminal**
2. **ip nbar protocol-pack-auto-update**
3. **source-server** *protocol-pack-auto-update-server*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.

Step 2	**ip nbar protocol-pack-auto-update**
	Example:
	`Device(config)#ip nbar protocol-pack-auto-update`
	`Device(config-auto-pp-update)#`
		Enables NBAR protocol pack auto update.

Step 3	**source-server** *protocol-pack-auto-update-server* **Example:** `Device(config-auto-pp-update)#source-server tftp://10.20.300.400/NbarAutoUpdate`	(Required only if the Protocol Pack Auto Update server has not already been specified) Specifies the location of the Protocol Pack Auto Update server and the directory containing the configuration file, NBAR_PROTOCOL_PACK_DETAILS.json.
Step 4	**exit** **Example:** `Device(config-auto-pp-update)#exit`	Exits global configuration mode.

Disabling Protocol Pack Auto Update

Disables Protocol Pack Auto Update on a router.

SUMMARY STEPS

1. **configure terminal**
2. **no ip protocol-pack-auto-update**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**no ip protocol-pack-auto-update** **Example:** `Device(config)# no ip nbar protocol-pack-auto-update`	Disables NBAR protocol pack auto update.
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Initiating Immediate Protocol Pack Update

Initiates an immediate Protocol Pack update using the Protocol Pack Auto Update mechanism.

SUMMARY STEPS

1. **configure terminal**
2. **ip nbar protocol-pack-auto-update now**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ip nbar protocol-pack-auto-update now** **Example:** `Device(config)# ip nbar protocol-pack-auto-update now`	Initiates a protocol pack update using the auto update mechanism.
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Displaying Protocol Pack Auto Update Information

Displays the Protocol Pack Auto Update configuration, copied files, and statistics for an individual router using Protocol Pack Auto Update.

SUMMARY STEPS

1. **show ip nbar protocol-pack auto-update**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip nbar protocol-pack auto-update** **Example:** `Device# show ip nbar protocol-pack-auto-update`	Displays the protocol pack auto update configuration, copied files, and statistics.

The following example shows the information provided in the output of this command.

```
Device# show ip nbar protocol-pack-auto-update

NBAR Auto-Update:
=================

Configuration:
=============
force-upgrade               : (Default)  Enabled
clear-previous              : (Default)  Enabled
update-window               : (Default)  30
source-server               :                tftp://10.20.200.1/NbarAutoUpdate/
protocol-pack-directory     : (Default)  harddisk:
schedule                    : (Default)  03:22

Copied files:
==========
File            : harddisk:/NbarAutoUpdate/AsrNbarPP
Copied          : *11:29:11.000 UTC Mon Jan 5 2015


Last run result: SUCCESS
Last auto-update run                    : *11:29:12.000 UTC Mon Jan 5 2015
```

```
Last auto-update success                 : *11:29:12.000 UTC Mon Jan 5 2015
Last auto-update successful update       : *11:29:12.000 UTC Mon Jan 5 2015

Last auto-update server-config update    : *16:15:13.000 UTC Mon Jan 5 2015
Success count                             : 3
Failure count                             : 0
Success rate                              : 100 percent

Next AU maintenance estimated to run at  : *17:15:13.000 UTC Mon Jan 5 2015
Next AU update estimated to run at       : *03:41:00.000 UTC Tue Jan 6 2015
```

Configuring Local Protocol Pack Auto Update Settings on a Router

To configure local Protocol Pack Auto Update settings on a router, use the command sub-mode described here. Configuring local settings on the router overrides any settings specified in the Protocol Pack Auto Update Configuration File.

SUMMARY STEPS

1. **configure terminal**
2. **ip nbar protocol-pack-auto-update**
3. Use one or more of the Protocol Pack Auto Update sub-mode commands to configure local settings on the router.
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.

Step 2	**ip nbar protocol-pack-auto-update**
	Example:
	`Device(config)#ip nbar protocol-pack-auto-update`
	`Device(config-auto-pp-update)#`
		Enters Protocol Pack Auto Update configuration sub-mode, indicated by a change in the prompt to include "(config-auto-pp-update)".

Step 3	Use one or more of the Protocol Pack Auto Update sub-mode commands to configure local settings on the router.	See Protocol Pack Auto Update Sub-mode Commands, on page 81.
Step 4	**exit** **Example:** `Device(config-auto-pp-update)#exit`	Exit the command sub-mode.

Protocol Pack Auto Update Sub-mode Commands

Protocol Pack Auto Update sub-mode commands configure local Auto Update settings on a router. For information on entering the command sub-mode, see Configuring Local Protocol Pack Auto Update Settings on a Router, on page 80.

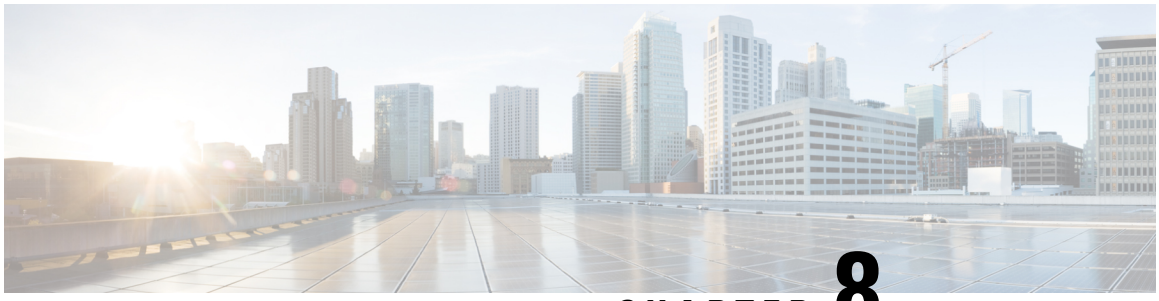
Use **exit** when finished to exit the command sub-mode.

Command	Description
clear-previous {**enable** | **disable**}	**enable**: Causes unneeded Protocol Pack files to be removed after a cool-down period. **disable**: Configures the feature to not remove any files. Default: Enable
force-upgrade {**enable** | **disable**}	**enable**: New Protocol Pack updates will be applied with the "force" flag. **disable**: New Protocol Pack updates will not be applied with the "force" flag. Default: Disable
protocol-pack-directory *directory*	Local directory in which to save new Protocol Pack files. Default: File system with highest space availability
schedule {**daily** | **weekly** | **monthly**} [*day*] [*hh:mm*]	Schedule the NBAR2 Protocol Pack Auto Update upgrade interval. The actual run time depends on the **update-window** option. Default: Daily at 00:00
update-window *minutes*	Maintenance window (in minutes) for NBAR2 Protocol Pack Auto Update to operate within. The maintenance window occurs according to the time configured by the **schedule** option. Range: 0 to 60 Default: 60

Example: Overriding Update Window

The following command sets the update window to 10 minutes, overriding the setting specified in the Protocol Pack Auto Update configuration file.

```
Device# configure terminal
Device(config)# ip nbar protocol-pack-auto-update
Device(config-auto-pp-update)# update-window 10
```



CHAPTER 8

Creating a Custom Protocol

Network-Based Application Recognition (NBAR) recognizes and classifies network traffic on the basis of a set of protocols and application types. You can add to the set of protocols and application types that NBAR recognizes by creating custom protocols.

Creating custom protocols is an optional process. However, custom protocols extend the capability of NBAR to classify and monitor additional static port applications and allow NBAR to classify nonsupported static port traffic.

This module contains concepts and tasks for creating a custom protocol.

Prerequisites for Creating a Custom Protocol

Before creating a custom protocol, read the information in the "Classifying Network Traffic Using NBAR" module.

Information About Creating a Custom Protocol

NBAR and Custom Protocols

NBAR supports the use of custom protocols to identify custom applications. Custom protocols support static port-based protocols and applications that NBAR does not currently support.

Note For a list of NBAR-supported protocols, see the "Classifying Network Traffic Using NBAR" module.

With NBAR supporting the use of custom protocols, NBAR can map static TCP and UDP port numbers to the custom protocols.

Initially, NBAR included the following features related to custom protocols and applications:

- Custom protocols had to be named custom-xx, with xx being a number.
- Ten custom applications can be assigned using NBAR, and each custom application can have up to 16 TCP and 16 UDP ports each mapped to the individual custom protocol. The real-time statistics of each custom protocol can be monitored using Protocol Discovery.

NBAR includes the following characteristics related to user-defined custom protocols and applications:

- The ability to inspect the payload for certain matching string patterns at a specific offset.
- The ability to allow users to define the names of their custom protocol applications. The user-named protocol can then be used by Protocol Discovery, the Protocol Discovery MIB, the **match protocol** command, and the **ip nbar port-map** command as an NBAR-supported protocol.
- The ability of NBAR to inspect the custom protocols specified by traffic direction (that is, traffic heading toward a source or a destination rather than traffic in both directions).
- CLI support that allows a user configuring a custom application to specify a range of ports rather than specify each port individually.
- The **http**/**dns**/**ssl** keyword group that lets you add custom host and URL signatures.

Note Defining a user-defined custom protocol restarts the NBAR feature, whereas defining predefined custom protocol does not restart the NBAR feature.

MQC and NBAR Custom Protocols

NBAR recognizes and classifies network traffic by protocol or application. You can extend the set of protocols and applications that NBAR recognizes by creating a custom protocol. Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allow NBAR to classify nonsupported static port traffic. You define a custom protocol by using the keywords and arguments of the **ip nbar custom** command. However, after you define the custom protocol, you must create a traffic class and configure a traffic policy (policy map) to use the custom protocol when NBAR classifies traffic. To create traffic classes and configure traffic polices, use the functionality of the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC). The MQC is a command-line interface that allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach these traffic policies to interfaces. For more information about NBAR and the functionality of the MQC, see the "Configuring NBAR Using the MQC" module.

Limitations of Custom Protocols

The following limitations apply to custom protocols:

- NBAR supports a maximum of 120 custom protocols. All custom protocols are included in this maximum, including single-signature and composite protocols.
- Cannot define two custom protocols for the same target regular expression.

For example, after configuring **ip nbar custom 1abcd http url www.abcdef.com**, cannot then configure:

```
ip nbar custom 2abcd http url www.abcdef.com
```

Attempting to do so results in an error.

- Maximum length for the regular expression that defines the custom protocol: 30 characters

How to Create a Custom Protocol

Defining a Custom NBAR Protocol Based on a Single Network Protocol

Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allow NBAR to classify non-supported static port traffic.

This procedure creates a custom NBAR protocol based on a single network protocol (HTTP, SSL, and so on).

Note NBAR supports a maximum of 120 custom protocols. All custom protocols are included in this maximum, including single-signature and composite protocols.

To define a custom protocol, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *protocol-name* [*offset* [*format value*]] [**variable** *field-name field-length*] [*source* | *destination*] [**tcp** | **udp**] [**range** *start end* | *port-number*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *protocol-name* [*offset* [*format value*]] [**variable** *field-name field-length*] [*source*	*destination*] [**tcp**	**udp**] [**range** *start end*	*port-number*]	Extends the capability of NBAR Protocol Discovery to classify and monitor additional static port applications or allows NBAR to classify non-supported static port traffic.

	Command or Action	Purpose
	Example: `Router(config)# ip nbar custom app_sales1 5 ascii SALES source tcp 4567`	• Creates a custom NBAR protocol that identifies traffic based on a single network protocol. • Useful for identifying a single type of traffic (HTTP, SSL, and so on) according to a specified pattern. • Enter the custom protocol name and any other optional keywords and arguments.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Examples

In the following example, the custom protocol LAYER4CUSTOM will look for TCP packets that have a destination or source port of 6700:

```
Device# configure terminal
Device(config)# ip nbar custom LAYER4CUSTOM  transport tcp id 14
Device(config-custom)# port 6700
```

To display other options besides port:

```
Device# configure terminal
Device(config)# ip nbar custom LAYER4CUSTOM  transport tcp id 14
Device(config-custom)# ?
Custom protocol commands:
  direction  Flow direction
  dscp       DSCP in IPv4 and IPv6 packets
  exit       Exit from custom configuration mode
  ip         ip address
  ipv6       ipv6 address
  no         Negate a command or set its defaults
  port       ports
```

Defining a Custom NBAR Protocol Based on Multiple Network Protocols

Custom protocols extend the capability of NBAR Protocol Discovery to classify and monitor additional static port applications and allow NBAR to classify non-supported static port traffic.

This procedure creates a custom NBAR protocol based on multiple network protocols.

Note In this description, the term "protocol" is used in two ways: as an NBAR protocol used for identifying traffic, and as a network protocol (HTTP, SSL, and so on).

Note NBAR supports a maximum of 120 custom protocols. All custom protocols are included in this maximum, including single-signature and composite protocols.

To define a composite-signature custom protocol, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *protocol-name* **composite server-name** *server-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *protocol-name* **composite server-name** *server-name* **Example:** `Router(config)# ip nbar custom abc_example_custom composite server-name *abc_example`	Extends the capability of NBAR Protocol Discovery to classify and monitor additional static port applications or allows NBAR to classify non-supported static port traffic. • Creates a custom NBAR protocol that identifies traffic using signatures for multiple network protocols. Currently, the only option for *composite-option* is **server-name**, which identifies all HTTP, SSL, and DNS traffic associated with a specific server. • Useful for identifying multiple types of traffic (HTTP, SSL, and so on) according to a specified pattern, using a single protocol. In the example, the objective is to identify all HTTP, SSL, and DNS traffic associated with the **abc_example.com** server.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Configuring a Traffic Class to Use the Custom Protocol

Traffic classes can be used to organize packets into groups on the basis of a user-specified criterion. For example, traffic classes can be configured to match packets on the basis of the protocol type or application recognized by NBAR. In this case, the traffic class is configured to match on the basis of the custom protocol.

To configure a traffic class to use the custom protocol, perform the following steps.

Note The **match protocol**command is shown at Step 4. For the *protocol-name* argument, enter the protocol name used as the match criteria. For a custom protocol, use the protocol specified by the *name* argument of the **ip nbar custom**command. (See Step 3 of the Defining a Custom Protocol task.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match protocol** *protocol-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all**	**match-any**] *class-map-name* **Example:** `Router(config)# class-map cmap1`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • Enter the name of the class map.
Step 4	**match protocol** *protocol-name* **Example:** `Router(config-cmap)# match protocol app_sales1`	Configures NBAR to match traffic on the basis of the specified protocol. • For the *protocol-name* argument, enter the protocol name used as the match criterion. For a custom protocol, use the protocol specified by the *name* argument of the **ip nbar custom**command. (See Step 3 of the "Defining a Custom Protocol" task.)

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-cmap)# end`	(Optional) Exits class-map configuration mode.

Examples

In the following example, the **variable** keyword is used while creating a custom protocol, and class maps are configured to classify different values within the variable field into different traffic classes. Specifically, in the example below, variable scid values 0x15, 0x21, and 0x27 will be classified into class map active-craft, while scid values 0x11, 0x22, and 0x25 will be classified into class map passive-craft.

```
Router(config)#
 ip nbar custom ftdd 23 variable scid 1 tcp range 5001 5005

Router(config)#
class-map active-craft
Router(config-cmap)# match protocol ftdd scid 0x15
Router(config-cmap)# match protocol ftdd scid 0x21
Router(config-cmap)# match protocol ftdd scid 0x27

Router(config)#
class-map passive-craft
Router(config-cmap)# match protocol ftdd scid 0x11
Router(config-cmap)# match protocol ftdd scid 0x22
Router(config-cmap)# match protocol ftdd scid 0x25
```

Configuring a Traffic Policy

Traffic that matches a user-specified criterion can be organized into specific classes. The traffic in those classes can, in turn, receive specific QoS treatment when that class is included in a policy map.

To configure a traffic policy, perform the following steps.

Note The **bandwidth** command is shown at Step 5. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Creates or modifies a policy map that can be attached to one or more interfaces and enters policy-map configuration mode. • Enter the name of the policy map.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change and enters policy-map class configuration mode. • Enter the specific class name or enter the **class-default**keyword.
Step 5	**bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*} **Example:** `Router(config-pmap-c)# bandwidth percent 50`	(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map. • Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth. **Note** The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	(Optional) Exits policy-map class configuration mode.

Attaching the Traffic Policy to an Interface

After a traffic policy (policy map) is created, the next step is to attach the policy map to an interface. Policy maps can be attached to either the input or output direction of the interface.

Note Depending on the needs of your network, you may need to attach the policy map to a subinterface, an ATM PVC, a Frame Relay DLCI, or other type of interface.

To attach the traffic policy to an interface, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface ethernet 2/4`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.
Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**| **qsaal**| **smds**| **l2transport**] **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 5	**exit**	(Optional) Returns to interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config-atm-vc)# exit`	**Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Step 4. If you are not attaching the policy map to an ATM PVC, advance to Step 6.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an input or output interface. • Enter the name of the policy map. **Note** Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached vary according to your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration.
Step 7	**end** **Example:** `Router(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Displaying Custom Protocol Information

After you create a custom protocol and match traffic on the basis of that custom protocol, you can use the **show ip nbar port-map** command to display information about that custom protocol.

To display custom protocol information, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show ip nbar port-map** [*protocol-name*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**show ip nbar port-map** [*protocol-name*] **Example:** `Router# show ip nbar port-map`	Displays the current protocol-to-port mappings in use by NBAR. • (Optional) Enter a specific protocol name.
Step 3	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Creating a Custom Protocol

Example Creating a Custom Protocol

In the following example, the custom protocol called app_sales1 identifies TCP packets that have a source port of 4567 and that contain the term SALES in the first payload packet:

```
Router> enable
Router# configure terminal
Router(config)# ip nbar custom app_sales1 5 ascii SALES source tcp 4567
Router(config)# end
```

Example Configuring a Traffic Class to Use the Custom Protocol

In the following example, a class called cmap1 has been configured. All traffic that matches the custom app_sales1 protocol will be placed in the cmap1 class.

```
Router> enable
Router# configure terminal
Router(config)# class-map cmap1
Router(config-cmap)# match protocol app_sales1
Router(config-cmap)# end
```

Example Configuring a Traffic Policy

In the following example, a traffic policy (policy map) called policy1 has been configured. Policy1 contains a class called class1, within which CBWFQ has been enabled.

```
Router> enable

Router# configure terminal

Router(config)# policy-map policy1

Router(config-pmap)# class class1

Router(config-pmap-c)# bandwidth percent 50

Router(config-pmap-c)# end
```

Note In the above example, the **bandwidth** command is used to enable Class-Based Weighted Fair Queuing (CBWFQ). CBWFQ is only an example of one QoS feature that can be applied in a traffic policy (policy map). Use the appropriate command for the QoS feature that you want to use.

Example Attaching the Traffic Policy to an Interface

In the following example, the traffic policy (policy map) called policy1 has been attached to ethernet interface 2/4 in the input direction of the interface.

```
Router> enable

Router# configure terminal

Router(config)# interface ethernet 2/4

Router(config-if)# service-policy input policy1

Router(config-if)# end
```

Example Displaying Custom Protocol Information

The following is sample output of the **show ip nbar port-map** command. This command displays the current protocol-to-port mappings in use by NBAR. Use the display to verify that these mappings are correct.

```
Router# show ip nbar port-map
port-map bgp      udp 179
port-map bgp      tcp 179
port-map cuseeme  udp 7648 7649
```

```
port-map cuseeme  tcp 7648 7649
port-map dhcp     udp 67 68
port-map dhcp     tcp 67 68
```

If the **ip nbar port-map** command has been used, the **show ip nbar port-map** command displays the ports assigned to the protocol.

If the **no ip nbar port-map**command has been used, the **show ip nbar port-map** command displays the default ports. To limit the display to a specific protocol, use the *protocol-name* argument of the **show ip nbar port-map** command.

Additional References

The following sections provide references related to creating a custom protocol.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC, traffic policies (policy maps), and traffic classes	"Applying QoS Features Using the MQC" module
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
Information about enabling Protocol Discovery	"Enabling Protocol Discovery" module
Configuring NBAR using the MQC	"Configuring NBAR Using the MQC" module
Adding application recognition modules (also known as PDLMs)	"Adding Application Recognition Modules" module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

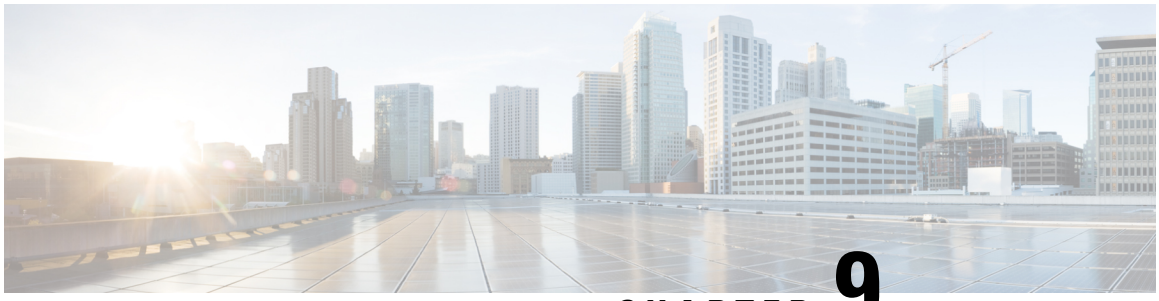
Feature Information for Creating a Custom Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for Creating a Custom Protocol

Feature Name	Releases	Feature Information
NBAR - Multiple Matches Per Port	12.4(2)T	Provides the ability for NBAR to distinguish between values of an attribute within the traffic stream of a particular application on a TCP or UDP port. The following sections provide information about the NBAR - Multiple Matches Per Port feature:
NBAR User-Defined Custom Application Classification	12.3(4)T	Provides ability to identify TCP- or UDP-based applications by using a character string or value. The character string or value is used to match traffic within the packet payload. The following sections provide information about the NBAR User-Defined Custom Application Classification feature:



CHAPTER 9

NBAR2 Custom Protocol

Network-Based Application Recognition (NBAR) supports the use of custom protocols to identify custom applications. Custom protocols support static port-based protocols and applications that NBAR does not support.

Prerequisites for NBAR2 Custom Protocol

Protocol pack licensing must be enabled to configure custom protocols.

Information About NBAR2 Custom Protocol

Overview of NBAR2 Custom Protocol

Network-Based Application Recognition (NBAR) supports the use of custom protocols to identify custom applications. Custom protocols support static port-based protocols and applications that NBAR does not support.

For more information about custom protocols, refer to "Creating a Custom Protocol" module.

IP Address and Port-based Custom Protocol

IP address and port-based custom protocol includes supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport. This enables Network-Based Application Recognition (NBAR) to recognize traffic based on IP addresses and to associate an application ID to traffic from and to specified IP addresses. You define a custom protocol transport by using the keywords and arguments of the **ip nbar custom transport** command.

To support the IP address and port-based custom protocol option, the custom configuration mode (config-custom) is introduced with the **ip nbar custom transport** command. This mode supports options to specify a maximum of eight individual IP addresses, subnet IP addresses, and subnet mask length. You can also specify a list of eight ports or a start port range and an end port range.

How to Configure NBAR2 Custom Protocol

Configuring IP Address and Port-based Custom Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *name* **transport {tcp | udp} {id** *id* **} {ip address** *ip-address* **| subnet** *subnet-ip subnet-mask***}| ipv6 address {***ipv6-address* **| subnet** *subnet-ipv6 ipv6-prefix***} | port {***port-number* **| range** *start-range end-range***} | direction {any | destination | source}**
4. **ip nbar custom** *name* **transport {tcp | udp} {id** *id***}**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip nbar custom** *name* **transport {tcp | udp} {id** *id* **} {ip address** *ip-address* **| subnet** *subnet-ip subnet-mask***}| ipv6 address {***ipv6-address* **| subnet** *subnet-ipv6 ipv6-prefix***} | port {***port-number* **| range** *start-range end-range***} | direction {any | destination | source}**
	Example:
	Specifies the IP address.
	`Device(config)# ip nbar custom mycustomprotocol transport tcp id 100`
	`Device(config-custom)# ip address 10.2.1.1`
	Example:
	Specifies the subnet IP and a subnet mask of 0.
		Configures the custom protocol, with options to specify IP address, subnet, port, direction, and so on. In the examples given, the command is executed on multiple lines, using the custom configuration mode, rather than the single-line format.

	Command or Action	Purpose
	`Device(config)# ip nbar custom mycustomprotocol transport tcp` `Device(config-custom)# ip subnet 255.255.255.255 0`	

Step 4	**ip nbar custom** *name* **transport {tcp | udp} {id** *id***}**
	Example:
	`Device(config)# ip nbar custom mycustom transport tcp id 100`
	`Device(config-custom)#`
		Specifies TCP or UDP as the transport protocol and enters custom configuration mode.

Step 5	**end** **Example:** `Device(config-custom)# end`	(Optional) Exits custom configuration mode.

Configuration Examples for NBAR2 Custom Protocol

Example: Configuring IP Address and Port-based Custom Protocol

The following example shows how to enter custom configuration mode from global configuration mode and configure a subnet IP address and its mask length:

```
Device(config)# ip nbar custom mycustomprotocol transport tcp id 100
Device(config-custom)# ip subnet 10.1.2.3 22
```

The following example configures two custom protocols, one for TCP and one for UDP traffic. In each, the subnet, subnet mask, DSCP value, and direction are configured.

```
Device(config)# ip nbar custom mycustomprotocol_tcp transport tcp
Device(config-custom)# ip subnet 255.255.255.255 0
Device(config-custom)# dscp 18
Device(config-custom)# direction any
Device(config-custom)# end
Device(config)# ip nbar custom mycustomprotocol_udp transport udp
Device(config-custom)# ip subnet 255.255.255.255 0
Device(config-custom)# dscp 18
Device(config-custom)# direction any
```

Additional References for NBAR2 Custom Protocol

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS LAN switching commands	Cisco IOS LAN Switching Command Reference
Cisco IOS QoS configuration information	*QoS Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

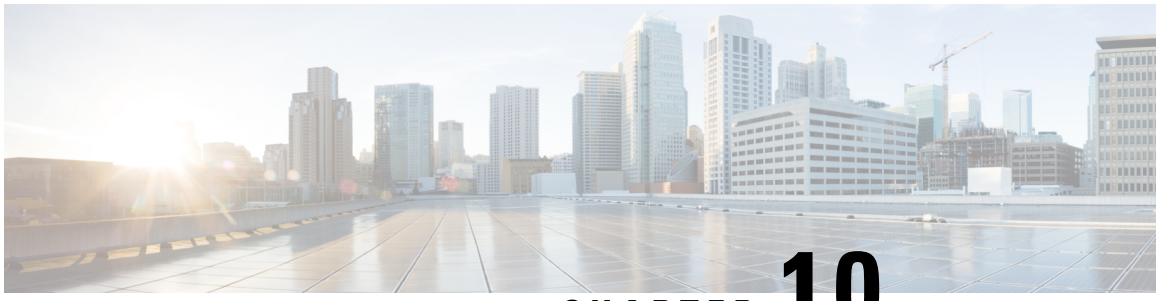
Feature Information for NBAR2 Custom Protocol

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for NBAR2 Custom Protocol

Feature Name	Releases	Feature Information
NBAR2 Custom Protocol Enhancements Ph II	15.4(2)T	The NBAR2 Custom Protocol Enhancements Phase II feature enables supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport. The following command was introduced or modified: **ip nbar custom**.



CHAPTER 10

NBAR Web-based Custom Protocols

The NBAR Web-based Custom Protocols feature provides the mechanism to define custom protocols to match based on HTTP URL and/or host name.

Restrictions for NBAR Web-based Custom Protocols

The HTTP URL and the Host name defined for custom protocol match should be unique. The length of the protocol name should be at least 4 characters long and the prefix of the protocol name should be different from the prefixes of any other protocol name.

Information About NBAR Web-based Custom Protocols

Overview of NBAR Web-based Custom Protocols

The NBAR Web-based Custom Protocols feature provides the mechanism to define custom protocols to match the traffic based on HTTP URL and/or host name.

All 120 custom protocols can be defined to match based on HTTP URL and/or host name. While matching web-based custom protocols, the custom protocol that has both HTTP URL and the host name defined has the highest priority, followed by HTTP URL as the second priority, and then followed by Host name as the last priority. Matching a web-based sub-protocol has higher priority than matching any type of web-based custom protocol, for example the **match protocol** *http url http-url* command has a higher priority than a custom priority with the same URL configuration.

How to Define NBAR Web-based Custom Protocols Match

Defining a Web-based Custom Protocol Match

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *custom-protocol-name* **http** {**host** *host-name* | **url** *http-url* [**host** *host-name*]} [**id** *selector-id*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *custom-protocol-name* **http** {**host** *host-name* | **url** *http-url* [**host** *host-name*]} [**id** *selector-id*] **Example:** `Router(config)# ip nbar custom app_sales1 http url www.example.com`	Defines web-based custom protocol match. • Enter the custom protocol name and any other optional keywords and arguments. **Note** To add a custom protocol, use the **ip nbar custom** command. To enable the protolcol, use the **match protocol** command or **ip nbar protocol discovery** command.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Configuration Examples for NBAR Web-based Custom Protocols

Examples: Defining Web-based Custom Protocol Match

The following example displays how to match a custom protocol based on http url:

```
Router> enable
Router# configure terminal
Router(config)# ip nbar custom app_sales1 http url www.example.com
```

The following example displays how to match a custom protocol that contains the string 'example' as a part of host name:

```
Router> enable
Router# configure terminal
Router(config)# ip nbar custom app_sales1 http host *example*
```

Additional References for NBAR Web-based Custom Protocols

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Custom Protocols	*Creating a Custom Protocol* module

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

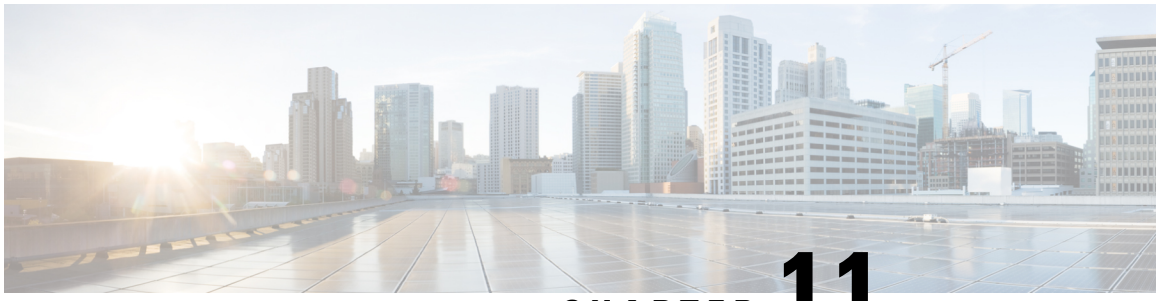
Feature Information for NBAR Web-based Custom Protocols

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for NBAR Web-based Custom Protocols

Feature Name	Releases	Feature Information
NBAR Web-based Custom Protocols Scalability	15.4(3)T 15.5(1)T	The NBAR Web-based Custom Protocols Scalability feature enables defining custom protocols match based on http host name and/or url. The following command was introduced or modified: **ip nbar custom**.



CHAPTER 11

NBAR2 HTTP-Based Visibility Dashboard

The NBAR2 HTTP-based Visibility Dashboard provides a web interface displaying network traffic data and related information. The information is presented in an intuitive, interactive graphical format.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Overview of NBAR2 HTTP-based Visibility Dashboard

The NBAR2 HTTP-based Visibility Dashboard provides a graphical display of network information, such as network traffic details and bandwidth utilization. The Visibility Dashboard includes interactive charts and a graph of bandwidth usage.

The basic workflow for using the Visibility Dashboard is:

1. Using the procedure described in Configuring NBAR2 HTTP-Based Visibility Dashboard, on page 107, configure the router to provide information for the Visibility Dashboard. This includes:
   - Enabling an HTTP server.
   - Setting up the router service that collects and stores traffic data.
   - Specifying an interface to monitor.

- Enabling protocol discovery.

2. In a browser, connect to the Visibility Dashboard web interface to display traffic information for the monitored interface(s), using the router IP address or hostname, and appending **/flash/nbar2/home.html**.

   Example: **10.56.1.1/flash/nbar2/home.html**

   See Accessing the Visibility Dashboard, on page 108.

3. The HTTP server that operates with the Visibility Dashboard requires HTTP command access to the router to collect traffic data to present in the dashboard. Specifically, the HTTP server executes **show ip nbar** CLI commands on the router to collect the data. Access is provided to the Visibility Dashboard HTTP server by one of the following methods:

   - Providing "privilege 15" general access to the router.

     Use the **ip http authentication enable** CLI command on the router to set a password. When logging into the Visibility Dashboard web interface, use the specified password. No username is required.

   - Setting a local username and password for the router.

     Use the **ip http authentication local** command to set a local username/password providing HTTP command access. When logging into the Visibility Dashboard web interface, enter the specified username and password.

     Example configuration:

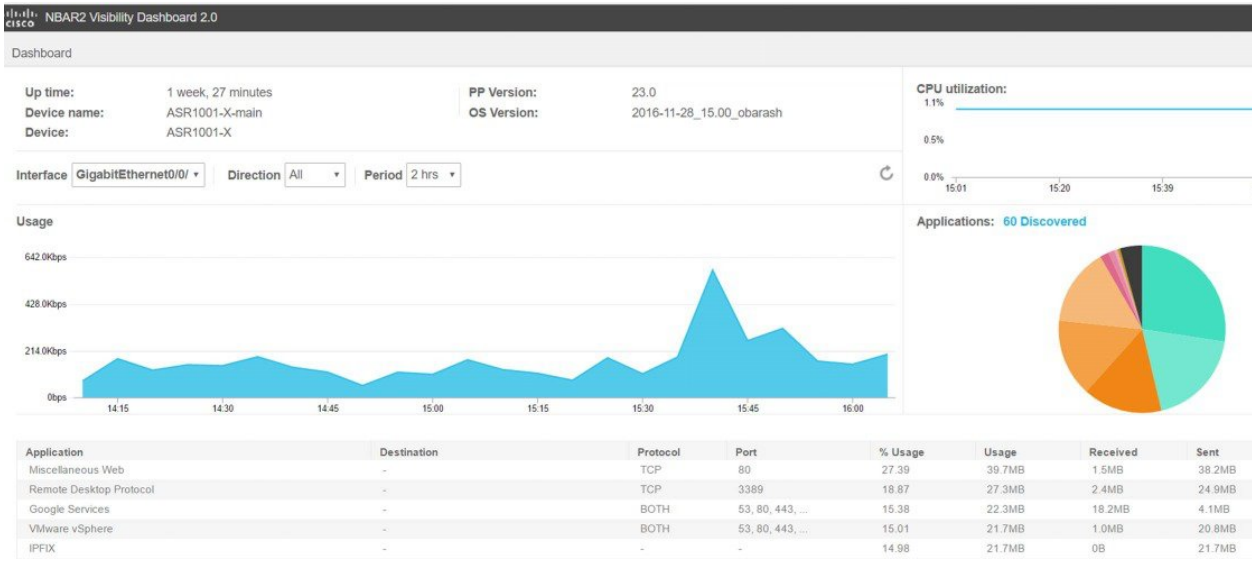
     ```
     Device(config)#ip http authentication enable
     Device(config)#ip http authentication local
     Device(config)#username cisco
     Device(config)#password n449rbpsvq
     ```

   - Using an Authentication, Authorization, and Accounting (AAA) server.

     The AAA server manages accounts, including username/password credentials. When logging into the Visibility Dashboard web interface, enter the username and password for an account managed by the AAA server.

     Note: The account must include authorization to execute **show ip nbar** commands on the router. If the account does not provide this authorization, a user could log in and pass authentication, but no traffic data would be available from the router. The Visibility Dashboard would appear in the browser, but showing no information.

Figure 2: Visibility Dashboard



Configuring NBAR2 HTTP-Based Visibility Dashboard

Before you begin

The HTTP-based Visibility Dashboard uses the Protocol Discovery feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip nbar http-services**
5. **interface gigabitethernet** *interface*
6. **ip nbar protocol-discovery**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter a password if prompted.
Step 2	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip http server** **Example:** `Device(config)#ip http server`	Enables an HTTP server. The server operates with the Visibility Dashboard, providing the data collected by the router.
Step 4	**ip nbar http-services** **Example:** `Device(config)#ip nbar http-services`	Configures the HTTP services to collect traffic data and store it in a database.
Step 5	**interface gigabitethernet** *interface* **Example:** `Device(config)#interface gigabitethernet 0/0/2`	Specifies an interface to monitor.
Step 6	**ip nbar protocol-discovery** **Example:** `Device(config)#ip nbar protocol-discovery`	

Example: NBAR2 HTTP-Based Visibility Dashboard

Example: Enabling NBAR2 HTTP-Services

```
Device> enable
Device# configure terminal
Device(config)# ip nbar http-services
Device(config)# end
```

Accessing the Visibility Dashboard

In a browser with access to the router, connect to the Visibility Dashboard web interface to display traffic information for the monitored interface(s), using the router IP address or hostname, and appending **/flash/nbar2/home.html**. This string is shown in the CLI help for **ip nbar http-services** by typing: **ip nbar ?**

Options:

- **http://<router-IP-address>/flash/nbar2/home.html**
- **http://<router-hostname>/flash/nbar2/home.html**

Example:

```
http://10.56.1.1/flash/nbar2/home.html
```

Additional References for NBAR2 HTTP-Based Visibility Dashboard

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

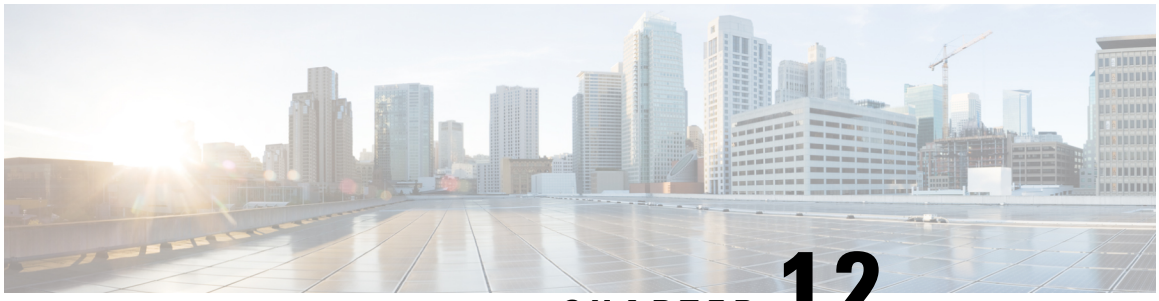
Feature Information for NBAR2 HTTP-Based Visibility Dashboard

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12: Feature Information for NBAR2 HTTP-Based Visibility Dashboard

Feature Name	Releases	Feature Information
NBAR2 HTTP-Based Visibility Dashboard	Cisco IOS XE Release 3.16S	The NBAR2 HTTP-based Visibility Dashboard provides a web interface displaying network traffic data and related information. The information is presented in an intuitive, interactive graphical format. The following command was modified or introduced by this feature: **ip nbar http-services**



CHAPTER 12

NBAR Coarse-Grain Classification

Information About NBAR Coarse-Grain Classification

Overview of NBAR Coarse-Grain Classification

NBAR provides two levels of application recognition-coarse-grain and fine-grain. By default NBAR operates in the coarse-grain mode.

By minimizing deep packet inspection, coarse-grain mode offers a performance advantage and reduces memory resource demands. This mode is useful in scenarios where the full power of fine-grain classification is not required.

Simplified Classification

Coarse-grain mode employs a simplified mode of classification, minimizing deep packet inspection. NBAR caches classification decisions made for earlier packets, then classifies later packets from the same server similarly.

Limitations of Coarse-Grain Mode

Coarse-grain mode has the following limitations in metric reporting detail:

- Granularity: Caching may result in some reduction in the granularity. For example, NBAR might classify some traffic as **ms-office-365** instead of as the more specific **ms-office-web-apps**.
- Evasive applications: Classification of evasive applications, such as BitTorrent, eMule, and Skype, may be less effective than in fine-grain mode. Consequently, blocking or throttling may not work as well for these applications.

Comparison of Fine-grain and Coarse-grain Modes

Coarse-grain mode has the following limitations in metric reporting detail:

	Fine-Grain Mode	Coarse-Grain Mode
Classification	Full-power of deep packet inspection	Simplified classification Some classification according to similar earlier packets.
Performance	Slower	Faster
Memory Resources	Higher memory demands	Lower memory demands
Sub-classification	Full supported	Partial support
Field Extraction	Full supported	Partial support
Ideal usage	Per-packet policy Example: class-map that looks for specific url	When there is no requirement for specific per-packet operations.

Additional References for NBAR Coarse-Grain Classification

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
AVC information	*AVC User Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	https://www.cisco.com/c/en/us/support/index.html

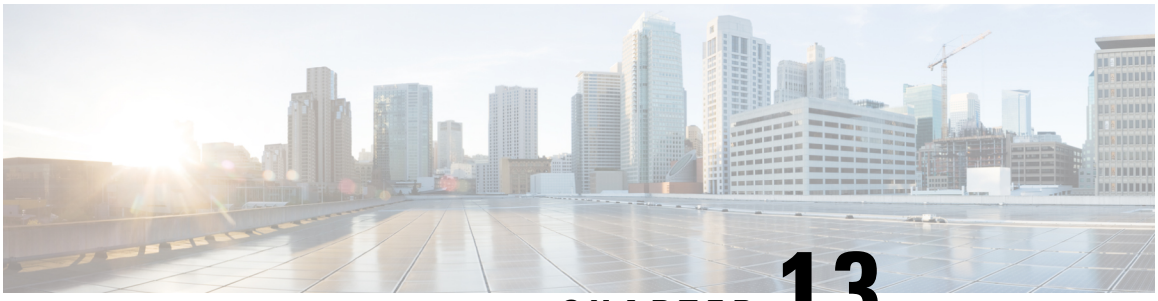
Feature Information for NBAR Coarse-Grain Classification

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for NBAR Coarse-Grain Classification

Feature Name	Releases	Feature Information
NBAR Coarse-Grain Classification		Network Based Application Recognition (NBAR) provides two levels of application recognition—coarse-grain and fine-grain. By default NBAR operates in the fine-grain mode, offering NBAR's full application recognition capabilities. By minimizing deep packet inspection, coarse-grain mode offers a performance advantage and reduces memory resource demands. The following command was introduced or modified: **ip nbar classification granularity** and **show ip nbar classification granularity**.
NBAR Coarse-Grain Classification	Cisco IOS XE Release 3.16S Cisco IOS XE 16.x releases	Default mode changed to coarse-grain.



CHAPTER 13

Fine-Grain NBAR for Select Applications

NBAR provides two levels of application recognition: coarse-grain and fine-grain modes. Coarse-grain mode optimizes performance. Fine-grain mode provides NBAR's full application recognition capabilities, but with a higher performance cost. By default, NBAR operates in coarse-grain mode.

Information About Fine-Grain NBAR for Selective Applications

Fine-Grain NBAR for Selective Applications

Overview

NBAR provides two levels of application recognition: coarse-grain and fine-grain modes. Coarse-grain mode optimizes performance. Fine-grain mode provides NBAR's full application recognition capabilities, but with a higher performance cost.

By default, NBAR operates in coarse-grain mode. NBAR automatically changes to fine-grain mode when required, based on the configuration and traffic patterns. Typically, it is not necessary to change NBAR's automatic behavior, but you can configure fine-grain mode manually, using the procedure described below.

Forcing fine-grain mode for specific applications may be useful for monitoring a subset of applications, without adversely affecting performance, while other applications continue in coarse-grain mode.

How to Configure Fine-Grain NBAR for Specific Applications

To override NBAR's automatic behavior and force fine-grain mode, use the following procedure. The procedure enables specifying applications individually by name or specifying applications that match a specific attribute value, such as "business-relevance = business-relevant".

Configure fine-grain mode:

```
enable
configure terminal
ip nbar classification granularity fine-grain { [protocol protocol-name] | [attribute
```

```
attribute-type attribute-value] }
exit
```

Display the currently configured NBAR classification mode:

```
show ip nbar classification granularity { [protocol protocol-name] | [attribute attribute-type
 attribute-value] }
```

Example

This example configures fine-grain mode for the application protocol, **cisco-media-audio**, then verifies with the **show** command.

```
Device#enable
Device#configuration terminal
Device(config)#ip nbar classification granularity fine-grain protocol cisco-media-audio
Device(config)#exit
Device#show ip nbar classification granularity protocol cisco-media-audio

Protocol                  Force mode
-------------------------------------
cisco-media-audio         fine-grain
```

How to Configure Fine-Grain NBAR for Selective Applications

Configuring Fine-Grain NBAR for Selective Applications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar classification granularity fine-grain protocol** *protocol-name*
4. **exit**
5. **show ip nbar classification granularity protocol** *protocol-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip nbar classification granularity fine-grain protocol** *protocol-name* **Example:** `Device(config)# ip nbar classification granularity fine-grain protocol 3pc`	Configures the fine-grain NBAR classification mode and specifies the protocol name which represents an application.
Step 4	**exit** **Example:** `Router(config)# exit`	Exits the global configuration mode and enters privileged EXEC mode.
Step 5	**show ip nbar classification granularity protocol** *protocol-name* **Example:** `Device(config)# show ip nbar classification granularity protocol 3pc`	Displays the currently configured NBAR classification mode.

Configuration Examples for Fine-Grained NBAR for Selective Applications

Example: Fine-Grain NBAR for Selective Applications

The following example shows how to configure the fine-grain classification mode of NBAR and select a protocol name that represents an application:

```
Device> enable
Device# configuration terminal
Device(config)# ip nbar classification granularity fine-grain protocol 3cp
Device(config)# exit
```

Example: Verifying the Fine-Grain NBAR for Selective Applications

The following example shows how to verify the classification granularity of the currently configured protocol:

```
Device # show ip nbar classification granularity protocol 3pc

Protocol                  Force mode
-------------------------------------
3pc                       fine-grain
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
AVC information	*AVC User Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	https://www.cisco.com/c/en/us/support/index.html

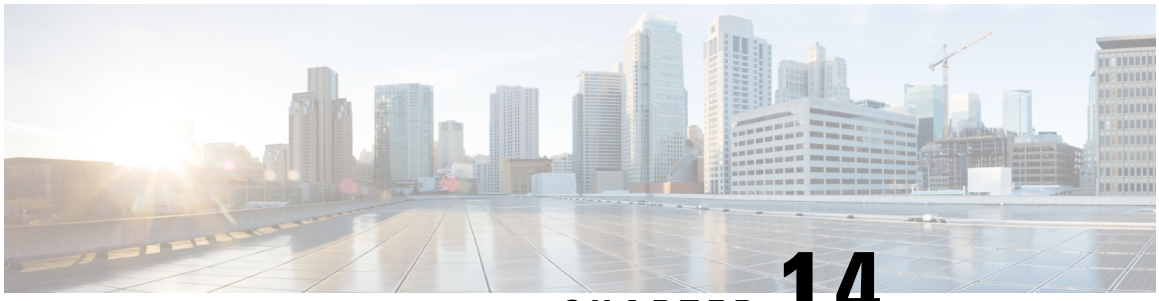
Feature Information for Fine-Grain NBAR for Selective Applications

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14: Feature Information for Fine-Grain NBAR for Selective Application

Feature Name	Releases	Feature Information
Fine-Grain NBAR for Selective Applications	15.5(2)T	By default NBAR operates in the fine-grain mode, offering NBAR's full application recognition capabilities. Used when per-packet reporting is required, fine-grain mode offers a troubleshooting advantage. Cisco recommends using fine-grain mode only when detailed Layer 7 metrics is required to be extracted by NBAR for critical applications. The fine-grain NBAR for Selective Apps feature enables a customer to dynamically monitor critical applications including collection of detailed Layer 7 metrics. The feature helps troubleshoot slowness in a particular application while the rest of the applications are running in in coarse-grain mode and thus preventing any impact on the performance of the system. The following command was introduced or modified: **ip nbar custom**.



CHAPTER 14

NBAR Custom Applications Based on DNS Name

NBAR Custom Applications based on DNS Name feature provides the mechanism to customize applications based on the Domain Name System (DNS) hostnames.

Prerequisites for NBAR Custom Applications Based on DNS Name

You must have basic knowledge of domain names.

Restrictions for NBAR Custom Applications Based on DNS Name

To use Domain Name System (DNS), you must have a DNS name server on your network.

DNS permits reading of UDP type messages only and considers only those response packets which have a source port of 53.

Information About NBAR Custom Applications Based on DNS Name

Overview of NBAR Custom Applications Based on DNS Name

Network-Based Application Recognition (NBAR) recognizes and classifies network traffic on the basis of a set of protocols and application types. The user adds to the set of protocols and application types that NBAR recognizes by creating custom protocols.

The user provides the DNS hostname signatures using the**ip nbar custom** *custom1 dns domain-name regular-expression id* command in the form of a simplified regular expression, which the DNS server pushes to the DNS templates. The DNS-based classification functions only when the IP addresses derived as direct responses are added to the look up table (LUT) for future classification lookups.

The following types of domains are supported:

- A
- AAAA
- CNAME

When you define the **ip nbar custom myDns dns domain-name** **example* command, the DNS traffic for a domain name that matches the expression "example" reaches the device. NBAR stores the corresponding IP address A.B.C.D of domain that matches the domain name with the expression "example" in its tables. When any TCP or UDP traffic with IP address A.B.C.D arrives, it is classified as myDns protocol.

How to Configure NBAR Custom Applications Based on DNS Name

Configuring the NBAR Custom Applications Based on DNS Name

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *custom-name* **dns** *domain-name regular-expression* **id** *1*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *custom-name* **dns** *domain-name regular-expression* **id** *1* **Example:** `Device(config)# ip nbar custom cust1 dns dns-name *example.com id 1`	Configures the NBAR Custom Applications Based on DNS Name feature. **Note** You can provide either the full domain name or a part of it as a regular expression. For example: the expression "*example" will match any domain that contains the word "example".
Step 4	**exit** **Example:** `Device(config)# exit`	Exits the global configuration mode and enters privileged EXEC mode.

Configuration Examples for NBAR Custom Applications Based on DNS Name

Example: Configuring NBAR Custom Applications Based on DNS Name

```
Device> enable
Device# configure terminal
Device(config)#  ip nbar custom custom1 dns domain-name *example id 11
Device(config)# exit
```

Additional References for NBAR Custom Applications Based on DNS Name

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

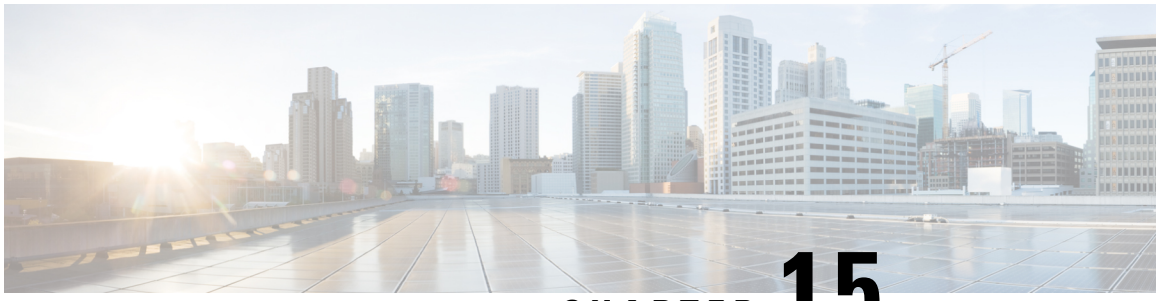
Feature Information for NBAR Custom Applications Based on DNS Name

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 15: Feature Information for NBAR Custom Applications Based on DNS Name

Feature Name	Releases	Feature Information
NBAR Custom Applications Based on DNS Name	15.5(2)T	NBAR custom applications based on Domain Name Service (DNS) Name feature provides the mechanism to customize applications based on the DNS hostnames. The following command was introduced or modified: **ip nbar custom**.



CHAPTER 15

DSCP-Based Layer 3 Custom Applications

Network-Based Application Recognition (NBAR) supports the use of custom protocols to identify customer-specific applications and applications that NBAR does not support. IP address and port-based custom protocol includes supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport. DSCP-Based Layer 3 Custom Applications feature is an enhancement that enables the customer to identify traffic that belongs to Layer 3 or Layer 4 custom applications by using Differentiated Services Code Point (DSCP) values in the traffic.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restriction of DSCP-Based Layer 3 Custom Applications

DSCP-Based Layer 3 Custom Applications feature treats the Differentiated Services Code Point (DSCP) classification as a property of the flow and checks only the DSCP value of the first packet in the flow. To identify different packets in the flow and apply policies on them, use the **match dscp** command.

DSCP-Based Layer 3 Custom Applications Overview

Network-Based Application Recognition (NBAR) supports the use of custom protocols to identify customer specific applications and applications that NBAR does not support. IP address and port-based custom protocol includes supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport. DSCP-based Layer 3 Custom Application feature is an enhancement that enables the customer to identify traffic that belongs to Layer 3 or Layer 4 custom applications by using Differentiated Services Code Point (DSCP) values in the traffic. You define a custom protocol transport by using the keywords and arguments of the **ip nbar custom transport** command.

How to Configure NBAR2 Auto-learn

Configuring DSCP-Based Layer 3 Custom Applications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *name* **transport** {**tcp** | **udp** | **udp-tcp** }**id** *id*
4. **dscp** *dscp-value*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip nbar custom** *name* **transport** {**tcp**	**udp**	**udp-tcp** }**id** *id* **Example:** `Device(config)# ip nbar custom mycustom transport tcp id 100`	Specifies the transport protocol to match as TCP, UDP, or both TCP and UDP, and enters custom configuration mode.
Step 4	**dscp** *dscp-value* **Example:**	Specifies the differentiated service code points (DSCP) value.

	Command or Action	Purpose
	`Device(config-custom)# dscp ef`	**Note** In cases where two custom applications have the same filters, the priority is set according to the order of configuration.
Step 5	**exit** **Example:** `Device(config-custom)# exit`	Exits custom configuration mode.

Configuration Examples for Configuring DSCP-Based Layer 3 Custom Applications

Example: DSCP-Based Layer 3 Custom Applications

```
Device> enable
Device# configuration terminal
Device (config)# ip nbar custom mycustom transport tcp id 100
Device(config-custom)# dscp ef
Device (config-custom)# exit
```

Example: L3 Custom any IP Port

```
Device> enable
Device# configuration terminal
Device (config)# ip nbar custom mycustom transport udp-tcp
Device(config-custom)# dscp ef
Device (config-custom)# exit
```

Additional References for DSCP-Based Layer 3 Custom Applications

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for DSCP-based Layer 3 Custom Applications

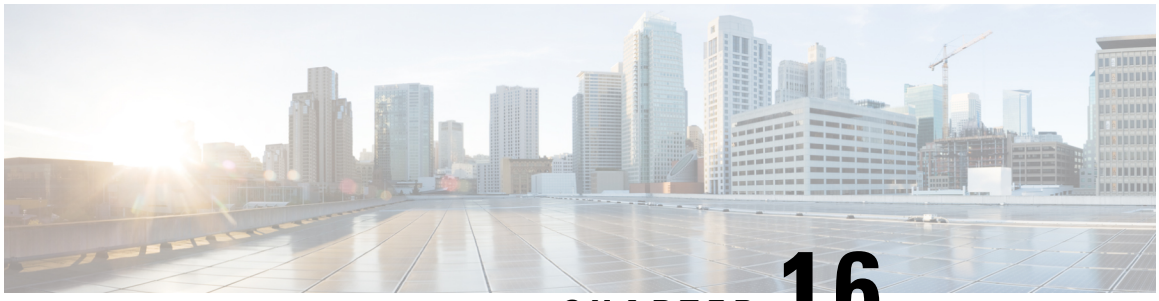
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 16:

Feature Name	Releases	Feature Information
DSCP-based Layer 3 Custom Applications	15.5(2)T, 15.5(3)T	NBAR supports the use of custom protocols to identify customer specific applications and applications that NBAR does not support. IP address and port-based custom protocol includes supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport. DSCP-based Layer 3 Custom Application feature is an enhancement that enables the customer to identify traffic that belongs to Layer 3 or Layer 4 custom applications by using DSCP values in the traffic. The following command was introduced or modified: **ip nbar custom**

Feature Name	Releases	Feature Information
L3 custom any IP/Port	Cisco IOS XE 3.16S	NBAR supports the use of custom protocols to identify customer specific applications and applications that NBAR does not support. IP address and port-based custom protocol includes supporting an IP subnet or a list of IP addresses with a specific TCP or UDP transport or TCP and UDP transport. DSCP-based Layer 3 Custom Application feature is an enhancement that enables the customer to identify traffic that belongs to Layer 3 or Layer 4 custom applications by using DSCP values in the traffic. The L3 Custom any IP/Port feature is an enhancement that enable users to to configure L3 or L4 custom applications over non UDP/TCP or over both UDP and TCP transport. The following command was introduced or modified: **ip nbar custom**



CHAPTER 16

NBAR2 Auto-learn

Important Beginning with Cisco IOS XE Fuji 16.9.1, this feature has been deprecated. The functionality has moved to Cisco Software-Defined AVC (SD-AVC).

NBAR2 Auto-learn improves classification of traffic not otherwise recognized by NBAR2 protocols. For generic HTTP or SSL traffic, NBAR2 can identify the hostname from packet header fields. For unknown traffic, it can track top-occurring server-side ports and sockets. These mechanisms facilitate creating custom protocols to better classify the otherwise generic or unknown traffic.

Note NBAR2 Auto-learn was previously called "NBAR Customized Assistance Based on SSL or HTTP."

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

NBAR2 Auto-learn Overview

A portion of network traffic may be difficult for NBAR2 mechanisms to identify specifically. Such traffic may be classified either as **generic** HTTP or SSL, or as **unknown**. This provides very little useful information about the traffic.

NBAR2 Auto-learn analyzes traffic classified as generic HTTP/SSL or unknown.

- For generic HTTP/SSL traffic, it derives hostnames from packet header fields in the traffic and tracks the "top hosts" that occur in generic traffic. This refers to the hosts with the highest traffic volume. The list of top hosts is arranged in order of traffic volume; hosts with the highest traffic volume are at the top of the list.
- For unknown traffic, it identifies server-side ports and tracks the "top ports" and "top sockets" that occur in unknown traffic. This refers to the ports and sockets with the highest traffic volume. The lists of top ports and sockets are arranged in order of traffic volume; ports and sockets with the highest traffic volume are at the top of the lists.

The lists of "top hosts" for generic and "top ports"/"top sockets" for unknown traffic can then be used to assist the custom protocol mechanism in creating protocols to better identify and classify the traffic. For example, top hosts provide "candidate" hosts to use in creating custom protocols.

Mechanism Details

NBAR supports the creation of custom protocols to identify traffic that built-in NBAR2 protocols do not recognize.

- For **generic** HTTP or SSL traffic, the NBAR2 Auto-learn can derive the relevant hostname from one of the following:
  - Server Name field in the Client Hello extensions
  - Common Name field in the digital certificate that a client sends to a server
- For **unknown** traffic, it can derive the server-side port number.

Example

For example, if NBAR2 is unable to classify traffic of an enterprise mail server, the traffic may be be classified only as SSL. This feature can assist in creating a custom protocol to identify the traffic more definitively, improving reporting of the mail server traffic.

How to Configure NBAR2 Auto-learn

Configuring NBAR2 Auto-learn

- For generic HTTP or SSL traffic, NBAR2 Auto-learn collects a list of the most often occurring hosts ("top hosts"). For unknown traffic, the feature collects a list of most often occurring server-side ports ("top ports") and sockets ("top sockets"). This information may be fed into the auto-custom mechanism to facilitate creating custom protocols.

- To optimize performance, the system does not track all flows of generic and unknown traffic. It samples flows using a specific sample rate. By default, for analyzing top hosts, NBAR2 sets the sample rate dynamically based on traffic. For information on configuring the sample rate, see Configuring NBAR2 Auto-learn, on page 132.
- By default, tracking top hosts is enabled; tracking top ports and top sockets is disabled.
- Auto-learn for "top sockets" is automatically enabled or disabled when "top ports" is enabled or disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar classification auto-learn** { **top-hosts** | **top-ports** }
4. **ip nbar classification auto-learn** { **top-hosts** | **top-ports** } **sample-rate** *rate*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter a password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**ip nbar classification auto-learn** { **top-hosts**	**top-ports** }
		Example:
		`Device (config)# ip nbar classification auto-learn top-hosts`
		`Device (config)# ip nbar classification auto-learn top-ports`
			top-hosts: Enables analyzing traffic classified as generic to generate a list of top hosts for the generic traffic.
			top-ports: Enables analyzing traffic classified as unknown to generate a list of server-side top ports and top sockets occurring in the unknown traffic.

Step 4	**ip nbar classification auto-learn** { **top-hosts**	**top-ports** } **sample-rate** *rate* **Example:** `Device (config)# ip nbar classification auto-learn top-ports sample-rate 5`	(Optional) Sets the flow sampling rate for the feature. To optimize performance, the mechanism does not track all generic and unknown traffic. It samples flows using a specific sample-rate. A smaller number improves accuracy, but requires more router resources. A *rate* value of 1 means that the mechanism samples all flows of generic (for **top-hosts**) or unknown (for **top-ports**) traffic. **top-hosts** default: NBAR2 sets the rate dynamically based on traffic.

	Command or Action	Purpose
		top-ports default: 128
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Displaying Auto-learn Top Hosts or Ports

SUMMARY STEPS

1. **show ip nbar classification auto-learn** { **top-hosts** | **top-ports** } *number_of_entries* [**detailed**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ip nbar classification auto-learn** { **top-hosts** | **top-ports** } *number_of_entries* [**detailed**]
	Example:
	`Device (config)# show ip nbar classification auto-learn top-hosts 10 detailed`
	
	`Device (config)# show ip nbar classification auto-learn top-ports 25`
		Displays statistics for the top hosts in generic traffic or top server-side ports occurring in unknown traffic.
		number_of_entries: Maximum number of entries to display. Possible values: 1 to 100
		detailed: Provides additional information, such as the byte, flow, and packet counts for each.

Displaying Auto-learn Top Sockets

In the context of auto-learn, sockets refer to server-side socket addresses (IP address and port).

Note The auto-learn top-sockets functionality is enabled or disabled automatically when top-ports is enabled or disabled.

SUMMARY STEPS

1. **show ip nbar classification auto-learn top-sockets** *number_of_entries* [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip nbar classification auto-learn top-sockets** *number_of_entries* [**detailed**] **Example:**	Displays statistics for the top sockets in unknown traffic. *number_of_entries*: Maximum number of entries to display. Possible values: 1 to 100

	Command or Action	Purpose
	`Device (config)# show ip nbar classification auto-learn top-sockets 100 detailed`	**detailed**: Provides additional information, such as the byte, flow, and packet counts for each.

Clearing Host/Port Statistics for NBAR2 Auto-learn

This procedure operates on the list of hosts, ports, and sockets that the NBAR2 Auto-learn feature creates for traffic classified as generic or unknown.

This command clears the statistical data (bytes, packets, flows, and so on) collected for the hosts (**top-hosts** option) or ports and sockets (**top-ports** option), but does not clear old hosts/ports/sockets for which no recent traffic has been detected. Compare this with **clear ip nbar classification auto-learn top-hosts restart**, which clears the statistics and also clears old hosts/ports/sockets.

SUMMARY STEPS

1. **clear ip nbar classification auto-learn** { **top-hosts** | **top-ports** } **statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip nbar classification auto-learn** { **top-hosts** | **top-ports** } **statistics** **Example:** `Device# clear ip nbar classification auto-learn top-hosts statistics`	Clears the statistical data collected for hosts (**top-hosts** option), or ports and sockets (**top-ports** option).

Clearing Host/Port Statistics and Inactive Hosts/Ports for NBAR2 Auto-learn

This procedure operates on the list of hosts, ports, and sockets that the NBAR2 Auto-learn feature creates for traffic classified as generic or unknown.

The procedure clears the statistical data (bytes, packets, flows, and so on) collected for the hosts (**top-hosts** option), or ports and sockets (**top-ports** option), and also clears the old hosts/ports/sockets for which no recent traffic has been detected. Compare this with **clear ip nbar classification auto-learn top-hosts statistics**, which clears the statistics, but does not clear old hosts/ports/sockets.

SUMMARY STEPS

1. **clear ip nbar classification auto-learn** { **top-hosts** | **top-ports** } **restart**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip nbar classification auto-learn** { **top-hosts** | **top-ports** } **restart** **Example:**	Clears the statistical data collected for hosts (**top-hosts** option) or ports (**top-ports** option), and also clears the old hosts/ports/sockets for which no recent traffic has been detected.

	Command or Action	Purpose
	`Device# clear ip nbar classification auto-learn top-hosts restart`	

Configuration Examples for NBAR2 Auto-learn

Example: Configuring Auto-learn for Hosts

```
Device> enable
Device# configuration terminal
Device (config)# ip nbar classification auto-learn top-hosts
Device (config)# exit
```

Example: Displaying Auto-learn Data

Top Hosts

Output of **show ip nbar classification auto-learn top-hosts** command without **detailed** option:

```
Device#show ip nbar classification auto-learn top-hosts 10

Total bytes:         23.236 M
Total packets:       31.816 K
Total flows:         229
Sample rate last:    1
Sample rate average: 1
Sample rate min:     1
Sample rate max:     1
-------------------------------------------------------------------------------
  #|Host                                |Byte%|Flow%|Pkt% |Type |Field
-------------------------------------------------------------------------------
  1|images1.xyz.com                     | 37% | 34% | 38% |http |host
  2|res.cloudinary.com                  | 34% |  3% | 25% |http |host
  3|mail.cisco.com                      | 27% | 62% | 35% |ssl  |host
  4|10.210.20.19                        | <1% | <1% | <1% |http |host
```

Top Hosts - Detailed

Output of **show ip nbar classification auto-learn top-hosts** command with **detailed** option:

```
Device# show ip nbar classification auto-learn top-hosts 10 detailed
Total bytes:         23.236 M
Total packets:       31.816 K
Total flows:         229
Sample rate last:    1
Sample rate average: 1
Sample rate min:     1
Sample rate max:     1
-----------------------------------------------------------------------------------------------

  #|Host                |Byte count |Byte%|Flow count |Flow%|Pkt count  |Pkt% |Type |Field
-----------------------------------------------------------------------------------------------
```

```
 1|site.xyz.com          |8.707 M    | 37% |79         | 34% |12.239 K   | 38% |http |host
 2|res.cloudinary.com    |8.045 M    | 34% |7          |  3% |8.162 K    | 25% |http |host
 3|mail.cisco.com        |6.363 M    | 27% |142        | 62% |11.315 K   | 35% |ssl  |host
 4|10.210.20.19          |120.111 K  | <1% |1          | <1% |100        | <1% |http |host
```

Top Sockets

In the context of auto-learn, sockets refer to server-side socket addresses (IP address and port).

Note The auto-learn top-sockets functionality is enabled or disabled automatically when top-ports is enabled or disabled.

Output of **show ip nbar classification auto-learn top-sockets** command (modified to fit more clearly):

```
Device#show ip nbar classification auto-learn top-sockets 100 detailed
Total bytes:  398.747 K
Total packets:         1.611 K
Total flows:           1.109 K
Sample rate last:    1
Sample rate average: 1
Sample rate min:     1
Sample rate max:     1
-------------------------------------------------------------------------------------------
 #|Port  |IP              |Byte count |Byte%|Flow |Flow%|Pkt   |Pkt% |Traffic |Asymmetric
  |      |                |           |     |count|     |count |     |Type    |byte
  |      |                |           |     |     |     |      |     |        |count
-------------------------------------------------------------------------------------------
 1|80    |173.38.201.172  | 81.776 K  | 20% |  4  | <1% | 90   |  5% |TCP     |0
 2|80    |173.38.201.174  | 74.555 K  | 18% |  4  | <1% | 84   |  5% |TCP     |0
 3|123   |10.56.129.33    | 42.672 K  | 10% |889  | 80% |889   | 55% |UDP     |N/A
 4|443   |47.88.68.98     |  1.472 K  | <1% |  3  | <1% | 10   | <1% |TCP     |0
 5|1080  |10.56.217.8     |  1     K  | <1% |  1  | <1% |  1   | <1% |TCP     |0
 6|63699 |10.210.20.123   | 213       | <1% |  1  | <1% |  1   | <1% |TCP     |0
 7|443   |171.70.124.118  | 37        | <1% |  1  | <1% |  1   | <1% |TCP     |0
 8|37814 |10.210.20.122   | 14        | <1% |  1  | <1% |  2   | <1% |TCP     |0
 9|443   |140.205.195.83  | 12        | <1% |  1  | <1% |  2   | <1% |TCP     |0
10|443   |10.61.25.91     |  7        | <1% |  1  | <1% |  1   | <1% |TCP     |0
```

Additional References for NBAR2 Auto-learn

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Solutions Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

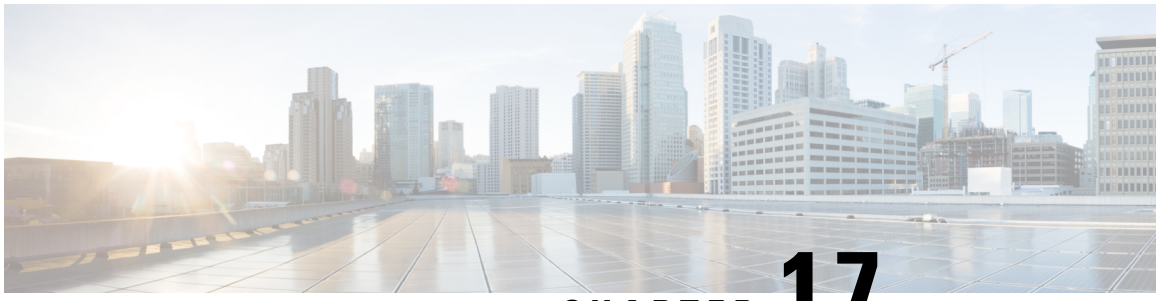
Feature Information for NBAR2 Auto-learn

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17: Feature Information for NBAR Customization Assistance Based on SSL or HTTP

Feature Name	Releases	Feature Information
NBAR2 Auto-learn (previously called "NBAR Customization Assistance based on SSL or HTTP")	Cisco IOS XE Release 3.16S, Cisco IOS Release 15.5(3)T	Assists in creating custom protocols to improve classification of generic or unknown traffic. The following commands were introduced or modified: **ip nbar classification auto-learn top-hosts**, **ip nbar classification auto-learn top-ports**, **ip nbar classification auto-learn top-ports sample-rate**, **show ip nbar classification auto-learn top-hosts**, **show ip nbar classification auto-learn top-ports**, **clear ip nbar classification auto-learn top-ports restart**, **clear ip nbar classification auto-learn top-hosts**, **clear ip nbar classification auto-learn top-ports statistics**



CHAPTER 17

Auto Traffic Analysis and Protocol Generation

NBAR includes an **auto-learn** feature that analyzes generic and unknown network traffic to determine the most frequently used hosts and ports. Using this data, the **auto-custom** feature can automatically generate NBAR protocols provisionally to improve identification of traffic.

Prerequisites for auto-custom

The auto-custom feature requires auto-learn to be active.

See NBAR2 auto-learn.

Limitations of auto-custom

Default

The auto-custom feature is disabled by default.

Environments Supported

The auto-custom feature supports the following environments:

- A single router with a single collector, or
- A single router with no collector

The feature does not support environments with multiple routers operating with a single collector.

Background: Auto Traffic Analysis Using NBAR2 Auto-learn

The NBAR2 **auto-learn** (see NBAR2 Auto-learn) and **auto-custom** features work together. NBAR2 Auto-learn analyzes traffic classified as generic HTTP/SSL or unknown. For generic HTTP/SSL traffic, it derives hostnames from packet header fields in the traffic and tracks the "top hosts" that occur in generic traffic. For unknown traffic, it identifies server-side ports and tracks the "top ports" and "top sockets" that occur in unknown traffic.

The results produced by **auto-learn** can be used by the **auto-custom** feature to automatically create custom NBAR protocols that improve classification of the traffic to improve application visibility for this difficult-to-classify traffic. For example, top hosts provide "candidate" hosts to use in creating custom protocols.

Auto Generation of Custom Protocols Using auto-custom

The **auto-custom** feature uses the results of **auto-learn** to improve NBAR classification of generic and unknown network traffic, automatically generating custom NBAR protocols.

Format for Reporting of Traffic Classified by Auto-generated NBAR Protocols

Auto-generated NBAR protocols report traffic according to hostname or port number:

- For **generic** traffic, protocols are generated for the most frequently occurring **hosts**, and are named according to the hostname. For traffic that contains only a host address and not a hostname, where possible, NBAR uses DNS lookup to provide the corresponding hostname.

  Examples: abcd.com, efgh.net

- For **unknown** traffic, protocols are generated for the most frequently occurring ports, and are named according to the **port number or socket** (server-side IP + port), and the traffic type: TCP or UDP.

  Examples for port: Port_80_TCP, Port_443_UDP

  Example for socket: 72.163.4.162:256_TCP

Auto-generation Is Based on Sampling of Traffic Flows

The **auto-learn** mechanism collects data about generic and unknown traffic by sampling traffic flows for analysis. Not every flow is analyzed. Using sampling rather than analyzing each flow is necessary due to the constraints of hardware resources. The availability of hardware resources for auto-learn analysis depends mostly on the network traffic volume that a device is handling.

For **generic** traffic, the sampling rate is dynamic, adjusting automatically according to system load. For **unknown** traffic, the default sampling rate is 128, meaning that the mechanism samples 1 flow for every 128 of unknown traffic. This value can be configured manually.

Because the **auto-custom** feature relies on data collected by **auto-learn**, the flow sampling performed by auto-learn can influence the automatic generation of protocols by auto-custom. In most use cases, however, sampling accurately reflects the makeup of network traffic.

Use of Auto-generated NBAR Protocols By Other Features

The NBAR application protocols auto-generated by auto-custom improve network traffic reporting, improving application visibility. However, the auto-generated protocols present at any given time are determined by the makeup of recent network traffic, making them inherently dynamic and impermanent.

Because of this dynamic nature, auto-custom protocols are applicable to some features, but not to others. In general, auto-custom protocols improve application **visibility**, but do not affect **security** (firewall) or **QoS** policies.

Features affected by auto-custom protocols:

- NBAR protocol discovery
- Application visibility (FNF, performance-monitor, ezPM, MACE, ...)

Features not affected by auto-custom protocols:

- MQC/QoS
- WAAS
- Performance Routing (PfR)
- NAT

Enabling and Disabling auto-custom

Enables or disables one or both of the auto-custom modes:

- top-hosts
- top-ports

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip nbar auto-custom** {**top-ports** | **top-hosts**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	[**no**] **ip nbar auto-custom** {**top-ports**	**top-hosts**} **Example:** `Device(config)# ip nbar auto-custom top-hosts`	Enables or disables auto-custom. The **top-ports** and **top-hosts** options apply the command to those respective modes of auto-custom.

	Command or Action	Purpose
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Configuring the Maximum Number of Auto-generated NBAR Protocols to Create

Configures the maximum number of protocols automatically generated by **auto-custom**. The auto-generated protocols present at any given time are determined by the makeup of recent network traffic, making them inherently dynamic and impermanent.

SUMMARY STEPS

1. **configure terminal**
2. **ip nbar auto-custom** {**top-hosts** | **top-ports**} **max-protocols** *number*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip nbar auto-custom** {**top-hosts** | **top-ports**} **max-protocols** *number* **Example:** `ip nbar auto-custom top-hosts max-protocols 30`	Configures the maximum number of auto-custom protocols to generate from the lists of top-hosts or top-ports collected by the auto-learn mechanism. **top-hosts** default: 10 **top-ports** default: 10
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Configuring the Time Interval for Re-generating the auto-custom Protocols

Configures the time interval at which auto-custom reloads the lists of "top-hosts" for generic traffic and "top-ports" for unknown data. The lists are provided by the **auto-learn** mechanism. After reloading the lists, the **auto-custom** mechanism generates a new set of custom protocols based on the data, which reflects the

most recent network traffic. Because of this mechanism, the list of auto-custom protocols is dynamic, changing with the makeup of generic and unknown network traffic.

SUMMARY STEPS

1. **configure terminal**
2. **ip nbar auto-custom** {**top-hosts** | **top-ports**} **time-interval** *minutes*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip nbar auto-custom** {**top-hosts**	**top-ports**} **time-interval** *minutes* **Example:** `ip nbar auto-custom top-hosts time-interval 10`	Configures the time interval at which auto-custom reloads the lists of "top-hosts" for generic traffic and "top-ports" for unknown data. Default: 30 minutes
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Clearing auto-custom Data

SUMMARY STEPS

1. **configure terminal**
2. **clear ip nbar auto-custom** {**top-hosts** | **top-ports**} {**stats** | **restart**}
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**clear ip nbar auto-custom** {**top-hosts**	**top-ports**} {**stats**	**restart**} **Example:** `clear ip nbar auto-custom top-ports restart`	Clears auto-custom data. The **top-ports** and **top-hosts** options apply the command to those respective modes of auto-custom. **stats**: Clears only counters

	Command or Action	Purpose
		restart: Clears counters and removes all current auto-custom protocols.
Step 3	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.

Displaying Auto-generated NBAR Protocols Created by auto-custom

SUMMARY STEPS

1. **show ip nbar auto-custom** [**top-hosts** | **top-ports**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip nbar auto-custom** [**top-hosts**	**top-ports**] **Example:** `show ip nbar auto-custom`	Displays the auto-generated NBAR protocols created by the auto-custom mechanism. Optionally, can specify only protocols for **top-hosts** or **top-ports**. • The first part of the output shows the protocols based on hostnames, from **generic** traffic. • The second part of the output shows the protocols based on port numbers + traffic type (TCP or UDP), from **unknown** traffic.

Example

```
# show ip nbar auto-custom
Top-hosts:
 Max number of protocols   :10
 Interval (min)            :30
 ------------------------------------------------------------------------------------------------
 | Id  | Protocol name          | Underlying | Auto-learn value               | Age (min) | Status |
 |     |                        | protocol   |                                |           |        |
 ------------------------------------------------------------------------------------------------
 |    1|m.abc-demo.com          |http        |m.abc-demo.com                  |         80|Dynamic |
 |    2|hwcdn.def-demo.com      |http        |hwcdn.def-demo.com              |         80|Dynamic |
 |    3|ec.def-demo.com         |http        |ec.def-demo.com                 |         80|Dynamic |
 |    4|payroll.demo.com        |ssl         |payroll.demo.com                |         80|Dynamic |
 |    5|ec-media.demo.com       |http        |ec-media.demo.com               |         50|Dynamic |
 |    6|TrustedSourceServer_IMQ |ssl         |TrustedSourceServer_IMQA01      |         20|Dynamic |
 |    7|go.microsoft.com        |http        |go.microsoft.com                |         20|Dynamic |
 |    8|ping.chartbeat.net      |http        |ping.chartbeat.net              |         20|Dynamic |
 ------------------------------------------------------------------------------------------------

Top-ports:
 Max number of protocols   :40
 Interval (min)            :1
 ------------------------------------------------------------------------------------------------
 | Id  | Protocol name          |             Auto-learn value                  | Age (min) | Status |
 ------------------------------------------------------------------------------------------------
 |    1|Port_256_TCP            |Port_256_TCP                                   |          0|Dynamic |
```

```
|    2|72.163.4.162:256_TCP    |72.163.4.162:256_TCP                          |          0|Dynamic |
-----------------------------------------------------------------------------------------------------
```

Displaying NBAR Protocol Discovery Information for auto-custom Protocols

SUMMARY STEPS

1. **show ip nbar protocol-discovery stat auto-custom**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip nbar protocol-discovery stat auto-custom** **Example:** `show ip nbar protocol-discovery stats auto-custom`	Displays the auto-custom protocol discovery statistics.

Example

```
# show ip nbar protocol-discovery stats auto-custom

 Ethernet0/0

 Last clearing of "show ip nbar protocol-discovery" counters 1d05h


                            Input                    Output
                            -----                    ------
   ------------------------ ------------------------ ------------------------
www.abcdef-demo.com          152         0
Total                        152                      0
```