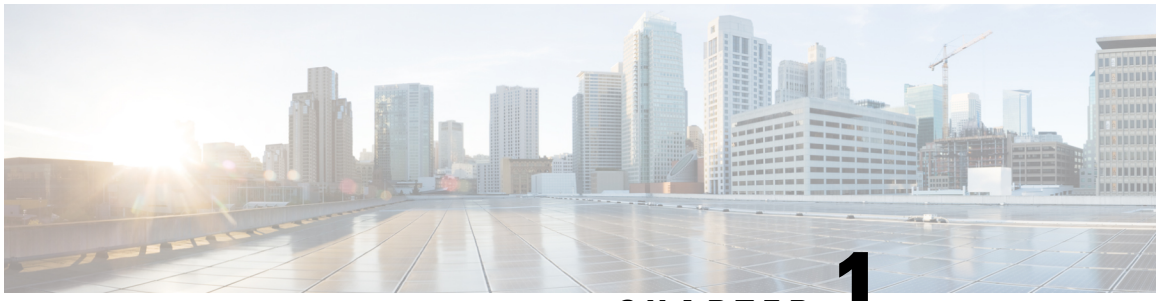
QoS: NBAR Configuration Guide, Cisco IOS XE Fuji 16.8.x

Last Modified: 2019-02-06

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

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

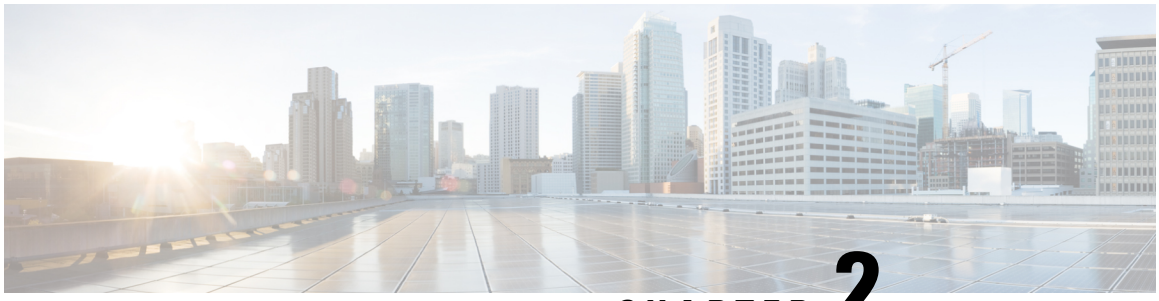
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.



CHAPTER 2

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 an application, you can configure the network to apply the appropriate quality of service (QoS) for that application or traffic with the classified protocol.

This module contains an overview of classifying network traffic using NBAR.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Classifying Network Traffic Using NBAR

NBAR does not support the following applications:

- Non-IP traffic.

- Multiprotocol Label Switching (MPLS)-labeled packets. NBAR classifies only IP packets. You can, however, use NBAR to classify IP traffic before the traffic is handed over to MPLS. Use the modular QoS CLI (MQC) to set the IP differentiated services code point (DSCP) field on NBAR-classified packets and make MPLS map the DSCP setting to the MPLS experimental (EXP) setting inside the MPLS header.
- NBAR processing. By design, NBAR processing is temporarily disabled during the In-Service Software Upgrade (ISSU). The following syslog message indicates the restart of the NBAR classification once ISSU is complete: "%NBAR_HA-5-NBAR_INFO: NBAR sync DONE!."
- Multicast packet classification.
- Asymmetric flows with stateful protocols.
- Packets that originate from or destined to a device running NBAR.

Note In the NBAR context, asymmetric flows are flows in which different packets go through different devices, for reasons such as load balancing implementation or asymmetric routing, where packets flow through different routes in different directions.

NBAR is not supported on the following logical interfaces:

- Dialer interfaces
- Dynamic tunnels such as Dynamic Virtual Tunnel Interface (DVTI)
- Fast Etherchannels
- IPv6 tunnels that terminate on the device
- MPLS
- Overlay Transport Virtualization (OTV) overlay interfaces

Note In cases where encapsulation is not supported by NBAR on some links, you can apply NBAR on other interfaces of the device to perform input classification. For example, you can configure NBAR on LAN interfaces to classify output traffic on the WAN link.

The following virtual interfaces are supported depending on the image of your Cisco IOS:

- Generic routing encapsulation (GRE)
- IPsec IPv4 tunnel (including tunneled IPv6) in protocol discovery mode and MQC mode
- IPsec IPv6 tunnel in protocol discovery mode but not in MQC mode
- Multipoint GRE/Dynamic Multipoint VPN (DMVPN) in protocol discovery mode

Note NBAR requires more CPU power when NBAR is enabled on tunneled interfaces.

If protocol discovery is enabled on both the tunnel interface and the physical interface on which the tunnel interface is configured, the packets that are designated to the tunnel interface are counted on both interfaces. On the physical interface, the packets are classified and are counted based on the encapsulation. On the tunnel interface, packets are classified and are counted based on the Layer 7 protocol.

For all protocols, only 20 combinations of subclassification per protocol can be configured. You can define a combination for subclassification using the **match protocol** *protocol-name variable-field-name value* command.

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 an 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 MQC.

Note For more information about the MQC, see the "Applying QoS Features Using the MQC" module.

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, that is, classification for inspecting packets.

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 includes the Protocol Pack feature that provides an easy way to load protocols and helps NBAR recognize additional protocols for network traffic classification. A protocol pack is set a of protocols developed and packed together. A new protocol pack can be loaded on the device to replace the default IOS protocol pack that is already present in the device.

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 network resources for the network traffic.

NBAR is also used in Cisco Application Visibility and Control (AVC). With AVC, NBAR provides better application performance through better QoS and policing, and provides finer visibility about the network that is being used.

With AVC license, the following NBAR features are supported:

- Classification inside transient IPv6 tunnels
- Custom protocols
- Customization of protocol attributes
- Field extraction
- Protocol pack updates

NBAR and Classification of HTTP Traffic

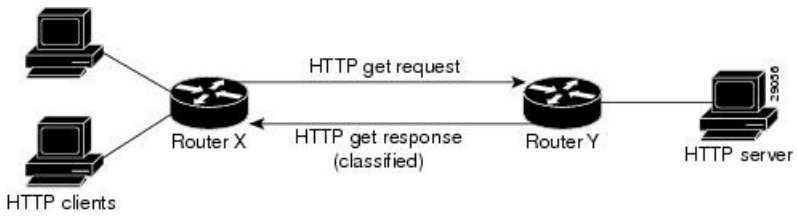
Classification of HTTP Traffic by a URL Host or MIME

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

Classification of HTTP traffic by a URL, a host, or a Multipurpose Internet Mail Extension (MIME) type is an example of subport classification. NBAR classifies HTTP traffic by the text within the URL or host fields of a request by 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 Device Y is the NBAR-enabled device.

Figure 1: Network Topology with an NBAR-enabled Device



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**).

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 not supported with subclassification and tunneled protocols that use HTTP as the transport protocol.

The NBAR Extended Inspection for HTTP Traffic 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.

Depending on your release, the Enable NBAR URI Extraction for HTTP Transactions for Persistent Connections feature supports extraction and export of the URL field per transaction, and not only the URL of the first transaction as supported in earlier releases. To enable multi-transaction, a protocol pack with 'Enhanced Web Classification' has to be installed. When an Enhanced Web Classification protocol pack is installed, the **match connection transaction-id** command configuration in flexible netflow tracks multiple HTTP transactions. For more information on tracking HTTP transactions, refer to *Cisco IOS Flexible NetFlow Configuration Guide*.

Note NBAR performs significant additional tasks for classification and export per transaction. These tasks impact performance and may cause increased export rate.

Classification of HTTP Traffic by Using HTTP Header Fields

NBAR introduces expanded ability for users to classify HTTP traffic by 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 is able to classify the following HTTP header fields:

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

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 The **c-header-field** and **s-header-field** keywords and associated arguments in the **match protocol http** command are no longer 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 *Quality of Service Solutions Command Reference*.

Note The c-header-field performs subclassifications based on a single value in the user-agent, the referrer, or from-header field values. The s-header-field performs subclassifications based on a single value 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 the 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 that is 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, use Server Browser Mode on the master browser.

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 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 in Published Desktop Mode. 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 is not able to differentiate among applications. Seamless sharing mode is enabled by default in some software releases. In seamless nonsession sharing mode, each application for each 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 a 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 would prefer printing to be handled as a background process that does not interfere with the processing of higher-priority traffic. To accommodate this printing 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 the Citrix client and server.

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 must 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 Quality of Service Solutions Command Reference.

The table below contains information about different Citrix traffic and the respective priority tags.

Table 1: Citrix ICA Packet Tagging

Priority	ICA Bits (decimal)	Sample Virtual Channels
High	0	Video, mouse, and keyboard screen updates
Medium	1	Program neighborhood, clipboard, audio mapping, and license management
Low	2	Client common equipment (COM) port mapping and client drive mapping
Background	3	Auto client update, client printer mapping, and original equipment manufacturers (OEM) channels

NBAR and RTP Payload Type Classification

Real-time Transport Protocol (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 part 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 and 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 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, does a deep-packet inspection into the RTP header to classify RTP packets.

For more information on the classification of RTP with NBAR, see NBAR RTP Payload Classification.

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 allow NBAR to classify nonsupported static port traffic.

Once the custom protocols are defined, you can then use them with the help of NBAR Protocol Discovery and the MQC to classify the traffic.

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

There are two types of custom protocols:

- Predefined custom protocols
- User-defined custom protocols

NBAR includes the following characteristics related to predefined custom protocols and applications:

- Custom protocols have 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.
- After creating a variable when creating a custom protocol, you can use the **match protocol** commandto classify traffic on the basis of a specific value in the custom protocol.

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, and the **match protocol**command as an NBAR-supported protocol.
- The ability of NBAR to inspect custom protocols specified by traffic direction (that is, traffic heading toward a source or destination rather than traffic in both directions), if desired by the user.
- CLI support that allows a user configuring a custom application to specify a range of ports rather than to specify each port individually.
- The **variable**keyword, the *field-name*argument,and the *field-length* argument were added to the **ip nbar custom**command.
- The **http** keyword group that lets you add custom host and URL signatures.

This additional keyword and two additional arguments allow for creation of more than one custom protocol based on the same port numbers.

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

NBAR DNS-based Classification

NBAR can improve traffic classification by using DNS transaction information exchanged when a user initiates a connection with an application server. This method offers the significant advantage of classifying flows from the first packet in the flow.

To illustrate, when a web-based application is opened in a browser, the browser first communicates with a DNS server to request the IP address of the relevant server for the application. The DNS transaction consists of a request and response; the response contains the IP address of the server for the web-based application.

Using information from this transaction, NBAR can correctly associate the web-based application with the relevant server IP address. NBAR can then identify future traffic involving that IP address from the first packet of the flow.

Supported Platforms

This feature is supported on platforms operating Cisco IOS XE, beginning with Cisco IOS XE release 3.17S, and including IOS XE Denali 16.x.

Advantages

NBAR applies multiple methods to classifying traffic, including in some cases, classifying traffic from the first packet, such as by socket-cache. The DNS-based classification feature operates with other NBAR methods to improve traffic classification. It is especially helpful for certain specific types of traffic, including asymmetric server-to-client flows, as well as some types of encrypted traffic.

Complementarity with Other NBAR Classification Methods

In general, the NBAR engine uses numerous strategies together to provide the most granular possible classification of traffic. First-packet classification may occur by multiple methods, including DNS-based classification and socket-cache. Additional classification methods may then add greater granularity to the classification.

Limitations

- Identification by DNS transaction information is insufficient in some situations. In these cases, NBAR relies on other methods to classify the traffic, where possible. For example, this method does not function well with generic hosts or service aggregation. (In the case of generic hosts or service aggregation, numerous services are hosted through a single server IP address, either using the same host name or different host names.)
- In some cases, NBAR may not be have access to the DNS transaction data for some traffic. For example, a network topology might include a local DNS server accessed through a connection not monitored by NBAR. DNS-based classification is not possible in these cases.

Limiting or Disabling DNS-based Classification

DNS-based classification may be disabled (see Enabling and Disabling DNS-based Classification, on page 20).

Typically, it is recommended to leave the DNS Guard feature in its default enabled state, which limits DNS-based Categorization to operating only when the complete DNS transaction (request, response) is available, but in special cases, it can be disabled (see Enabling and Disabling DNS Guard for DNS-based Categorization, on page 21).

Related Functionality

In addition to the DNS-based classification feature, NBAR has other methods that can, in some cases, provide first packet classification of traffic.

Customized server specification. This feature operates on all platforms that support NBAR, including those that do not support the DNS-based classification method. This feature is more limited than the DNS transaction method in its functionality. Customized server specification requires user configuration of the specific domains to identify using the DNS transaction information.

Use of customized server specification overrides other NBAR classification methods for the specified domain, and should only be used when specifically required. For information about this feature, including configuration commands, see: NBAR Custom Applications Based on DNS Name.

NBAR and Classification with Dynamic PDLMs

Dynamic Packet Description Language Modules (PDLMs) allow new protocol support or enhance existing protocol support for NBAR without the requirement of a specific Cisco release upgrade and device reload. If the support is for enhancing protocols for NBAR, the module version of the PDLMs should be greater than the existing version of the PDLMs. Subsequent Cisco releases incorporate support for these new protocols.

Note PDLMs must be loaded on both Route Processors (RPs) when using the ASR 1006 redundant hardware setup.

Dynamic PDLMs are platform-specific and have a Software Family Identifier (SFI) embedded in them. Dynamic PDLMs of other platforms cannot be loaded on Cisco ASR 1000 Series Aggregation Services Routers.

NBAR-Supported Protocols

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

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

NBAR2 Protocol Pack

The NBAR2 Protocol Pack provides an easy way to update protocols supported by NBAR2 without replacing the base IOS image that is already present in the device. A Protocol Pack is a set of protocols developed and packaged together. To view the list of protocols supported in a Protocol Pack, see NBAR2 Protocol Library.

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

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

- BitTorrent
- DirectConnect
- eDonkey
- eMule
- FastTrack
- KazaA (and KazaA Lite and KazaA Lite Resurrection)
- Win MX
- POCO

DirectConnect and eDonkey P2P protocols support the following subclassifications depending on your release:

- eDonkey supports the following subclassification options:
  - file-transfer
  - search-file-name
  - text-chat
- KazaA, FastTrack, and Gnuetella support the file-transfer subclassification.

The Gnutella file sharing became classifiable using NBAR in Cisco IOS XE Release 2.5.

Applications that use the Gnutella protocol are Bearshare, Gnewtellium, Gnucleus, Gtk-Gnutella, Limewire, Mutella, Phex, Qtella, Swapper, and Xolo. The traffic from the applications that use the Gnutella protocol will be classified as Gnutella and not as the respective application.

NBAR Multi stage Classification

NBAR supports a wide range of stateful network protocols such as HTTP classification by URL, Host and MIME type, FTP, TFTP, and so on. NBAR classifies static-port protocols such as those classifiable with access control lists (ACLs).

Multi stage classification reports the underlying protocol as a temporary classification instead of an unknown classification. For example, in earlier releases, to support cases like Video-over-HTTP, where the signature is found on the HTTP response packet, recursive classification over HTTP was allowed causing the first packet of HTTP flows to be reported as unknown, which in turn impacted the following:

- Protocol discovery—reduced classification.
- Packet-based flexible NetFlow (FNF)—reduced classification.

- QoS—delayed classification.
- Performance—because more packets were being processed.
- Aging short flows that are in the middle of a classification process stops without any classification results, although they were partially classified.

Prior to NBAR multi stage classification, NBAR reported an unknown classification result until a final classification decision was reached. NBAR multi stage classification returns the most up-to-date classification decision. It modifies the data path to expose the underlying protocols from media partitioning (MP) recursive classification path—instead of returning "unknown" until a final classification is available, it returns the current (temporary) classification decision.

NBAR multi stage classification has the following characteristics:

Backward incompatibility

If a system has a policy that matches a protocol like SOCKet Secure (SOCKS), which is an underlying protocol for AOL Instant Messenger (AIM) and Bittorrent, when all other protocols have failed (when other protocols are also enabled, either through protocol discovery or through FNF or explicitly through modular QoS CLI [MQC]), this policy would match the first packets of AIM or Bittorrent flows as SOCKS. Blocking the underlying protocol while allowing non underlying protocols is not possible with multi stage classification.

Traffic Reordering

When a user configures different priorities for each classification on the traffic flow, the flow might be directed to different output queues. With multi stage classification more than one classification decision for a single traffic flow may occur. When the traffic is based on prioritized classification, we recommend that the underlying protocols get a higher priority (for example, HTTP get a higher priority than Video-over-HTTP).

Performance Routing (PfR)

When PfR checks the classification from NBAR to make a routing decision, it takes into account if this is a final classification or not. If it is not the final classification, no routing decision is made as it may split the traffic flow to many paths resulting in an "unknown" classification.

NBAR clients let the users know if the classification is temporary or not.

NBAR Scalability

Interface Scalability

Depending on your release there is no limit to the number of interfaces on which protocol discovery can be enabled.

The following table provides details of the protocol discovery supported interface and the release number.

Table 2: Release and Protocol Discovery Interface Support

Release	Number of Interfaces Supported with Protocol Discovery
Cisco IOS XE Release 2.5	128
Cisco IOS XE Release 2.6	256
Cisco IOS XE Release 2.7	256

Release	Number of Interfaces Supported with Protocol Discovery
Cisco IOS XE Release 3.2S and later releases	256

Flow Scalability

The number of bidirectional flows and the platforms supported are same for all releases. A method to reduce the number of active flows based on quick aging is available.

Quick aging occurs under the following conditions:

- TCP flows that do not reach the established state.
- UDP flows with fewer than five packets that are not classified within the specified quick aging timeout.
- Flows that are not classified within the specified quick aging timeout.

The quick aging method reduces the number of flows required for NBAR operation up to three times or more depending on the network behavior.

The Cisco Cloud Services Router 1000V Series devices exhibit the same behavior as that of ESP5 with respect to flow scalability.

Flow Table Sizing

The **ip nbar resources flow max-sessions** command provides the option to override the default maximum flow sessions that are allowed in a flow table. The performance of the device with the NBAR feature depends on the memory size and the number of flows configured for the flow table. The flexibility to change the number of flows helps in increasing the performance of the system depending on the capacity of the device. To verify the NBAR flow statistics, use the **show ip nbar resources flow** command.

The following table provides the details of the platform and the flow size limits:

Table 3: Platform and Flow Size Details

Platform	Maximum Number of Flows	Default Number of Flows	Memory Upper Limit (70% of Platform Memory)
ESP5/ASR1001/CSR	750,000	500,000	179 MB
ESP10	1,650,000	1,000,000	358 MB
ESP20/ESP40/ASR1002-X	3,500,000	1,000,000	716 MB
ESP100	10,000,000	3,000,000	2.1 GB

To reduce the memory impact, the recommended number of flows is 50,000, where such a configuration is sufficient.

Note The total number of flow entries does not increase when the overall system memory usage is at or above 90%.

NBAR Protocol Discovery

NBAR includes a feature called Protocol Discovery. Protocol discovery provides an easy way to discover protocol packets passing through an interface. For more information about Protocol Discovery, see the "Enabling Protocol Discovery" module.

NBAR Protocol Discovery MIB

The NBAR Protocol Discovery MIB expands the capabilities of NBAR Protocol Discovery by providing the following new functionalities through the 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 the traffic of particular NBAR-supported protocols or applications that report breaches and send notifications when these thresholds are exceeded.

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

NBAR and Multipacket Classification

Depending on your release, NBAR provides the ability to simultaneously search large number of multipacket signatures. This new technique is supported for many of the new protocols. This technique also provides improved performance and accuracy for other protocols. Along with the support for new signatures, the multipacket classification capabilities change NBAR behavior in the following ways:

1. NBAR classification requires anywhere between 1 and 15 payload packets in a flow depending on the protocol. Retransmitted packets are not counted in this calculation.
2. NBAR will neither classify flows without any payload packets nor classify any TCP payload packet with a wrong sequence number even if there are 15 payload packets for classification.
3. TCP retransmitted packets are not counted as valid packets for classification in the Multipacket Engine module. These type of packets can delay the classification until a sufficient number of valid payload packets are accumulated.
4. Payload packets with only static signatures in NBAR are classified after the single-packet and multipacket protocols are processed and failed. Therefore, a maximum of 15 payload packets can be classified as unknown until the final (static) classification decision is taken.
5. Due to the above-mentioned restrictions, custom protocols can be used to force the classification of the first packet, ignoring the existence of payload or correct sequence numbers in the port-based classification.

NBAR on VRF Interfaces

Depending on your release, the NBAR IPv4 and IPv6 classification on VRF interfaces is supported.

Note Classification for Citrix protocol with “app” subclassification is not guaranteed on VRF interfaces when NBAR is enabled on VRF interfaces.

NBAR and IPv6

Depending on your release, the following types of classification are supported:

- NBAR provides static port-based classification and IP protocol-based classification for IPv6 packets.
- NBAR supports IPv6 classification in protocol discovery mode, but not in MQC mode.
- NBAR always reads the next header field in the fixed IPv6 header to determine the transport layer protocol used by the packet’s payload for IPv6 packets. If an IPv6 packet contains one or more extension headers, NBAR will not skip to the last IPv6 extension header to read the actual protocol type; instead, NBAR classifies the packet as an IPv6 extension header packet.

NBAR Support for IPv6

Depending on your release, NBAR supports the following types of classification:

- Native IPv6 classification.
- Classification of IPv6 traffic flows inside tunneled IPv6 over IPv4 and teredo.
- IPv6 classification in protocol discovery mode and in MQC mode.
- Static and stateful classification.
- Flexible NetFlow with NBAR based fields on IPv6.

NBAR supports IPv6 in IPv4 (6-to-4, 6rd, and ISATAP), and teredo tunneled classification. The **ip nbar classification tunneled-traffic** command is used to enable the tunneled traffic classification. When the tunneled traffic classification is enabled, NBAR performs an application classification of IPv6 packets that are carried inside the IPv4 traffic. If the **ip nbar classification tunneled-traffic** command is disabled, the tunneled IPv6 packets are handled as IPv4 packets.

NBAR supports the capture of IPv6 fields and allows the creation of IPv6 traffic-based flow monitors. When you enable the **ipv6 flow monitor** command, the monitor is bound to the interface, NBAR classification is applied to the IPv6 traffic type, and Flexible NetFlow captures the application IDs in the IPv6 traffic flow.

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.

NBAR Support for CAPWAP

CAPWAP (Control And Provisioning of Wireless Access Points) is a protocol is used in wireless traffic, providing point-to-point encapsulation (tunnel) for application traffic. There are two types of CAPWAP traffic: data and control.

NBAR provides a CAPWAP recognition mode that enables NBAR classification of the application traffic within a CAPWAP tunnel.

Classification Behavior: CAPWAP Recognition Disabled/Enabled

By default, CAPWAP recognition mode is not enabled. All CAPWAP traffic is reported as "capwap-data" or "capwap-control" without details about the application traffic within the tunnel.

When CAPWAP recognition is enabled:

- CAPWAP control traffic: NBAR reports as "capwap-control."
- CAPWAP data traffic: NBAR reports on the specific application traffic within the tunnel.

CAPWAP Traffic Type	NBAR CAPWAP Recognition Enabled	NBAR CAPWAP Recognition Disabled
Control traffic	NBAR reports traffic as "capwap-control"	NBAR reports traffic as "capwap-control"
Data traffic	NBAR reports application traffic within the CAPWAP tunnel	NBAR reports traffic as "capwap-data"

Requirements

The following are required for the NBAR recognition of application traffic within a CAPWAP tunnel:

- Cisco IOS XE platform
- Cisco IOS XE 3.17 or later
- NBAR enabled on the platform

Usage

The CAPWAP feature is disabled by default. Use the **ip nbar classification tunneled-traffic capwap** CLI to enable the feature. To disable, use **no ip nbar classification tunneled-traffic capwap**.

```
device# config terminal
device(config)# ip nbar classification tunneled-traffic capwap
```

NBAR Configuration Processes

You can configure NBAR in the following two ways:

- Configuring NBAR using MQC
- Enabling Protocol Discovery

For more information about the NBAR configuration, see the QoS: NBAR Configuration Guide.

Restarting NBAR

NBAR is restarted under the following circumstances.

- Custom protocol addition via CLI
- PDLM load
- RP switchover
- FP switchover
- Protocol pack installation
- Link-age change

Restart involves deactivating and reactivating NBAR. During this time, all packets are classified as ‘Unknown’ by NBAR. Once NBAR is reactivated, classification is activated.

Note Protocol Discovery statistics will be lost with RP Switchover.

How to Configure DNS-based Categorization

The following procedures describe how to configure NBAR DNS-based Categorization, including enabling/disabling the feature overall, and enabling/disabling DNS Guard.

For background information, see NBAR DNS-based Classification, on page 12.

Enabling and Disabling DNS-based Classification

NBAR2 employs a traffic analysis mechanism called DNS-based classification that learns the network addresses of applications by analyzing DNS query/response traffic. This enables NBAR to classify application traffic from the first packet of a flow, sometimes called "first in flow" (FIF). The mechanism, sometimes called DNS-based learning, applies to applications described by protocols in the NBAR2 Protocol Pack provided by Cisco.

The mechanism is enabled by default. Disabling the feature may be useful if the mechanism causes mis-classification of traffic. Use the **no** form of the command to disable.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip nbar classification dns learning**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**[no] ip nbar classification dns learning** **Example:** `Device(config)#no ip nbar classification dns learning`	Enables or disables the DNS-based classification mechanism. This example disables the feature. Default: enabled

Enabling and Disabling DNS Guard for DNS-based Categorization

The DNS-based Categorization mechanism analyzes DNS request/response traffic in order to learn the network addresses of applications. When successful, this enables NBAR to classify the application traffic from the first packet in a flow. In unusual situations, it may cause mis-classification. The feature is disabled by default. See Enabling and Disabling DNS-based Classification, on page 20.

In typical use, it is recommended to apply DNS-based Categorization only when the complete DNS transaction (request, response) is available, in order to prevent mis-classification of traffic. The DNS Guard feature enables this control.

- **Enabled**: DNS-based Categorization operates only when both the DNS request and response are available to analyze.
- **Disabled**: DNS-based Categorization does not require a DNS request, and uses only the DNS response to learn the network address of applications. Use the **no** form of the command to disable.

The mechanism is disabled by default.

SUMMARY STEPS

1. **configure terminal**
2. **[no] ip nbar classification dns learning guard**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**[no] ip nbar classification dns learning guard** **Example:** `Device(config)#no ip nbar classification dns learning guard`	Enables or disables DNS Guard. This example disables the feature. Default: disabled

How to Classify Network Traffic Using NBAR

NBAR provides two approaches to configuring attribute-based protocol matching:

- Grouping traffic into **categories and sub-categories** (see Configuring Attribute-based Protocol Match Using Categories and Sub-categories, on page 24)

  Useful for policy implementations that do not use SRND. A disadvantage of this method is that it can be difficult to keep track of the mapping between traffic and the categories and sub-categories defined within the policy.

- Using the Solution Reference Network Designs (**SRND**) model (see Configuring Attribute-based Protocol Match Using SRND, on page 25)

  Simplifies the configuration of SRND-based policies. Although the category/sub-category model can support SRND implementations, it is simpler and more efficient to use this model.

About Configuring Attribute-based Protocol Matching Using Categories

Useful for policy implementations that do not use SRND. A disadvantage of this method is that it can be difficult to keep track of the mapping between traffic and the categories and sub-categories defined within the policy. For information about the procedure, see Configuring Attribute-based Protocol Match Using Categories and Sub-categories, on page 24.

About Configuring Attribute-based Protocol Matching Using SRND

The NBAR category/sub-category model can support SRND implementations. However, beginning with the release of IOS 15.5(3)T and IOS XE 3.16S, for SRND policy implementations it is more efficient and recommended to use the SRND-specific model instead.

The SRND-specific model provides two attributes (**traffic-class** and **business-relevance**) to configure protocol matching for SRND-based policies. The attributes provided for operation with SRND-based policies are applicable only within the context of SRND implementations.

Background: SRND Policy Model

The Solution Reference Network Designs (SRND) policy model simplifies prioritization of traffic for QoS. It provides 12 classes that define traffic according to application. Each class of traffic can be directed to a specific QoS queue. Of these classes:

- 10 classes apply to business-relevant applications operating in 10 different recognized technologies, such as VoIP, video, conferencing, and so on.
- 1 class applies to business-relevant applications of unknown technology.
- 1 class applies to business-irrelevant applications.

Flexibility to Reclassify Applications

The 12 classes that NBAR provides for operating with the SRND model include default values appropriate for most enterprises. However, NBAR makes it easy to reclassify specific applications as business-relevant

or business-irrelevant, as necessary. (See example of reclassifying the Skype VoIP application: Example: SRND Configuration - Reclassifying an Application as Business-relevant, on page 32)

Attribute: traffic-class

The **traffic-class** attribute specifies the general category of the traffic, such as VoIP, video, conferencing, and so on. The The following table describes the 10 values for **traffic-class**.

Table 4: Values for traffic-class

Value	Description
voip-telephony	VoIP telephony (bearer-only) traffic
broadcast-video	Broadcast TV, live events, video surveillance
real-time-interactive	High-definition interactive video applications
multimedia-conferencing	Desktop software multimedia collaboration applications
multimedia-streaming	Video-on-Demand (VoD) streaming video
network-control	Network control plane traffic
signaling	Signaling traffic that supports IP voice and video telephony
ops-admin-mgmt	Network operations, administration, and management traffic
transactional-data	Interactive data applications
bulk-data	Non-interactive data applications

Attribute: business-relevance

The business-relevance attribute specifies whether the application is considered relevant to the business activity of the organization. The default values reflect typical usage and business relevance, but the values can be customized according to the specific requirements of an organization.

The following table describes the values for business-relevance.

Table 5: Values for business-relevance

Value	Description
business-relevant	Application critical for an organization's business activity
default	Application used for an organization's business activity
business-irrelevant	Application not relevant to an organization's business activity

Configuring Attribute-based Protocol Match Using Categories and Sub-categories

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**type**] [**match-all** | **match-any**] *class-map-name*
4. **match protocol attribute application-group** *application-group* [*application-name*]
5. **match protocol attribute category** *application-category* [*application-name*]
6. **match protocol attribute encrypted** {**encrypted-no** | **encrypted-unassigned** | **encrypted-yes**} [*application-name*]
7. **match protocol attribute sub-category** *application-category* [*application-name*]
8. **match protocol attribute tunnel** {**tunnel-no** | **tunnel-unassigned** | **tunnel-yes**} [*application-name*]
9. **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** [**type**] [**match-all**	**match-any**] *class-map-name* **Example:** `Device(config)# class-map cmap1`	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 attribute application-group** *application-group* [*application-name*] **Example:** `Device(config-cmap)# match protocol attribute application-group skype`	Configures the specified application group as the match criterion. • (Optional) Use the *application-name* argument to configure the application and not the application group as the match criterion. The configuration is saved as **match protocol** *application-name* instead of **match protocol attribute application-group** *application-group*.
Step 5	**match protocol attribute category** *application-category* [*application-name*] **Example:** `Device(config-cmap)# match protocol attribute category email`	Configures the specified category as the match criteria attribute. • (Optional) Use the *application-name* argument to configure a specific application, and not the application category, as the match criterion. The configuration is

	Command or Action	Purpose
		saved as **match protocol** *application-name* instead of **match protocol attribute category** *application-category*.
Step 6	**match protocol attribute encrypted** {**encrypted-no** | **encrypted-unassigned** | **encrypted-yes**} [*application-name*] **Example:** `Device(config-cmap)# match protocol attribute encrypted encrypted-yes`	Configures the specified encryption status as the match criterion. • (Optional) Use the *application-name* argument to configure application within the specified encrypted status as the match criterion. The configuration is saved as **match protocol** *application-name* instead of **match protocol attribute encrypted** {**encrypted-no** | **encrypted-unassigned** | **encrypted-yes**}.
Step 7	**match protocol attribute sub-category** *application-category* [*application-name*] **Example:** `Device(config-cmap)# match protocol attribute sub-category client-server`	Configures the specified sub-category as the match criteria attribute. • (Optional) Use the *application-name* argument to configure a specific application, and not the sub-category, as the match criterion. The configuration is saved as **match protocol** *application-name* instead of **match protocol attribute sub-category** *application-category*.
Step 8	**match protocol attribute tunnel** {**tunnel-no** | **tunnel-unassigned** | **tunnel-yes**} [*application-name*] **Example:** `Device(config-cmap)# match protocol attribute tunnel tunnel-yes`	Configures the specified encryption status as the match criterion. • (Optional) Use the *application-name* argument to configure a specific application within the specified tunneling status as the match criterion. The configuration is saved as **match protocol** *application-name* instead of **match protocol attribute tunnel** {**tunnel-no** | **tunnel-unassigned** | **tunnel-yes**}.
Step 9	**end** **Example:** `Device(config-cmap)# end`	Exits Qos class-map mode and returns to privileged EXEC mode.

Configuring Attribute-based Protocol Match Using SRND

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**type**] [**match-all** | **match-any**] *class-map-name*
4. **match protocol attribute traffic-class** *traffic-class-option*
5. **match protocol attribute business-relevance** *business-relevance-option*

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	**class-map** [**type**] [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map cmap1`	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 attribute traffic-class** *traffic-class-option* **Example:** `Device(config-cmap)# match protocol attribute traffic-class voip-telephony`	Configures the specified traffic class as the match criterion. • *traff-class-option* possible values: voip-telephony, broadcast-video, real-time-interactive, multimedia-conferencing, multimedia-streaming, network-control, signaling, ops-admin- mgmt, transactional-data, bulk-data
Step 5	**match protocol attribute business-relevance** *business-relevance-option* **Example:** `Device(config-cmap)# match protocol attribute business-relevance business-relevant`	Configures the specified category as the match criteria attribute. • *business-relevance-option* possible values: business-relevant, default, business-irrelevant
Step 6	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map mode and returns to privileged EXEC mode.

SRND Configuration: Typical Class-Map, Policy-Map

The following sections show a typical example of a class-map and policy-map for an SRND implementation. It illustrates how the **traffic-class** and **business-relevance** attributes address the 12-class SRND QoS model.

Class-map

```
class-map match-all VOICE
    match protocol attribute traffic-class voip-telephony
    match protocol attribute business-relevance business-relevant

class-map match-all BROADCAST-VIDEO
    match protocol attribute traffic-class broadcast-video
```

```
    match protocol attribute business-relevance business-relevant

class-map match-all INTERACTIVE-VIDEO
    match protocol attribute traffic-class real-time-interactive
    match protocol attribute business-relevance business-relevant

class-map match-all MULTIMEDIA-CONFERENCING
    match protocol attribute traffic-class multimedia-conferencing
    match protocol attribute business-relevance business-relevant

class-map match-all MULTIMEDIA-STREAMING
    match protocol attribute traffic-class multimedia-streaming
    match protocol attribute business-relevance business-relevant

class-map match-all SIGNALING
    match protocol attribute traffic-class signaling
    match protocol attribute business-relevance business-relevant

class-map match-all NETWORK-CONTROL
    match protocol attribute traffic-class network-control
     match protocol attribute business-relevance business-relevant

class-map match-all NETWORK-MANAGEMENT
    match protocol attribute traffic-class ops-admin-mgmt
    match protocol attribute business-relevance business-relevant

class-map match-all TRANSACTIONAL-DATA
    match protocol attribute traffic-class transactional-data
    match protocol attribute business-relevance business-relevant

class-map match-all BULK-DATA
    match protocol attribute traffic-class bulk-data
    match protocol attribute business-relevance business-relevant

class-map match-all SCAVENGER
    match protocol attribute business-relevance business-irrelevant
```

Policy-map

```
policy-map 12-cls-marking

class VOICE
    set dscp ef

class BROADCAST-VIDEO
    set dscp cs5

class INTERACTIVE-VIDEO
    set dscp cs4

class MULTIMEDIA-CONFERENCING
    set dscp af41

class MULTIMEDIA-STREAMING
    set dscp af31

class SIGNALING
    set dscp cs3

class NETWORK-CONTROL
    set dscp cs6

class NETWORK-MANAGEMENT
```

```
    set dscp cs2

class TRANSACTIONAL-DATA
    set dscp af21

class BULK-DATA
    set dscp af11

class SCAVENGER
    set dscp cs1

class class-default
    set dscp default
```

Configuration Examples for Classifying Network Traffic Using NBAR in Cisco Software

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.

```
Device(config)# class-map match-all class1
Device(config-cmap)# 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.

```
Device(config)# class-map match-all class2
Device(config-cmap)# 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.

```
Device(config)# class-map match-all class3
Device(config-cmap)# 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.

```
Device(config)# class-map match-all class4
Device(config-cmap)# 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.

```
Device(config)# class-map match-all class5
Device(config-cmap)# 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.

```
Device(config)# class-map match-all class6
Device(config-cmap)# 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 Custom Protocols and Applications

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
```

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

The **match protocol gnutella file-transfer** *regular-expression* and **match protocol fasttrack file-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:

```
Device(config)# class-map match-all nbar
Device(config-cmap)# match protocol fasttrack file-transfer "*"
```

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

```
Device(config)# class-map match-all nbar
Device(config-cmap)# 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 are classified into class map nbar:

```
Device(config)# class-map match-all nbar
Device(config-cmap)# match protocol gnutella file-transfer "*.mpeg"
```

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

```
Device(config)# class-map match-all nbar
Device(config-cmap)# match protocol gnutella file-transfer "*cisco*"
```

The same examples can be used for FastTrack traffic:

```
Device(config)# class-map match-all nbar
Device(config-cmap)# match protocol fasttrack file-transfer "*.mpeg"
```

or

```
Device(config)# class-map match-all nbar
Device(config-cmap)# match protocol fasttrack file-transfer "*cisco*"
```

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:

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

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

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

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

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

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

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

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

```
Device# configure terminal
Device(config)# class-map mygroup
Device(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:

```
Device# 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, epayment, 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:

```
Device# 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: SRND Configuration - Reclassifying an Application as Business-relevant

Skype is a consumer VoIP product typically not used in business. In SRND-specific protocol mapping, Skype is classified as business-irrelevant by default. However, some organizations may use Skype as a business-critical application. This examples shows how to reclassify Skype as business-relevant.

1. Show the current protocol attributes for Skype. The results indicate (in the last two lines) that Skype is classified as a voip-telephony technology, and is business-irrelevant.

   ```
   show ip nbar protocol-attribute skype
   encrypted           encrypted-yes
   tunnel              tunnel-no
   category            voice-and-video
   sub-category        consumer-multimedia-messaging
   application-group   skype-group
   p2p-technology      p2p-tech-yes
   traffic-class       voip-telephony
   business-relevance  business-irrelevant
   ```

   At this stage, Skype will be matched by the SCAVENGER class-map, which is part of the standard default SRND class-map configuration.

   ```
   class-map match-all SCAVENGER
       match protocol attribute business-relevance business-irrelevant
   ```

2. Change the value of business-relevance for Skype to business-relevant.

   ```
   ip nbar attribute-map demo
       attribute business-relevance business-relevant
   ip nbar attribute-set skype demo
   ```

   At this stage, Skype will be matched by the VOIP-TELEPHONY class-map, which is part of the standard default SRND class-map configuration.

```
class-map match-all VOIP-TELEPHONY
    match protocol attribute traffic-class voip-telephony
    match protocol attribute business-relevance business-relevant
```

3. Confirm that Skype is now classified as business-relevant. The new value appears on the last line of the following results.

```
show ip nbar protocol-attribute skype
encrypted           encrypted-yes
tunnel              tunnel-no
category            voice-and-video
sub-category        consumer-multimedia-messaging
application-group   skype-group
p2p-technology      p2p-tech-yes
traffic-class       voip-telephony
business-relevance  business-relevant
```

Example: Customizing a Built-in Protocol

Customizing an NBAR built-in protocol (provided by the Cisco Protocol Pack) to include an additional user-specified domain extends the scope of the built-in protocol. Any policy associated with the protocol will then apply to the user-specified domain also. The following example configures a customization called myOffice365, which extends the built-in office365 protocol to include domains that match to "*uniqueOffice365".

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

```
Device# configure terminal
Device(config)# ip nbar custom myOffice365 dns domain-name "*uniqueOffice365" extends
office365
```

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 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 6: Feature Information for Classifying Network Traffic Using NBAR

Feature Name	Releases	Feature Information
Additional PDL Support for NBAR	Cisco IOS XE Release 3.1S	The additional PDL Support for NBAR feature provides support for additional PDLs. The following section provides information about this feature: NBAR and Classification of HTTP Traffic
Enable NBAR URI Extraction for HTTP Transactions for Persistent Connections	Cisco IOS XE Release 3.9S	The Enable NBAR URI Extraction for HTTP Transactions for Persistent Connections feature supports extraction and export of URL field per transaction. The following section provides information about this feature: Classification of HTTP Traffic by a URL Host or MIME.
Enhanced NBAR	Cisco IOS XE Release 3.2S	The Enhanced NBAR feature provides additional PDLs for Cisco IOS XE Release 3.2S. The following section provides information about this feature: NBAR-Supported Protocols
NBAR Classification Enhancements for IOS-XE3.5	Cisco IOS XE Release 3.5S	The NBAR Classification Enhancements feature provides additional classification support for native IPv6 classification and classification of flows inside tunneled IPv6 over IPv4. The following section provides information about this feature: NBAR Support for IPv6 The following commands were introduced or modified: **ip nbar classification tunneled-traffic**, **option** (FNF).

Feature Name	Releases	Feature Information
NBAR PDLM Supported in ASR 1000 Release 2.5	Cisco IOS XE Release 2.5 Cisco IOS XE Release 3.1S Cisco IOS XE Release 3.3S	This feature was integrated into Cisco IOS XE Release 2.5. NBAR-supported protocols were added for this release. The following section provides information about this feature: NBAR-Supported Protocols The following command was modified: **match protocol** (NBAR).
NBAR Protocols	Cisco IOS XE Release 2.3	This feature was integrated into Cisco IOS XE Release 2.3. NBAR-supported protocols were added for this release. The following section provides information about this feature: NBAR-Supported Protocols The following command was modified: **match protocol**(NBAR).
NBAR Real-time Transport Protocol Payload Classification	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following section provides information about this feature: NBAR-Supported Protocols
NBAR Static IPv4 IANA Protocols Pack1	Cisco IOS XE Release 3.1S	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following section provides information about this feature: NBAR-Supported Protocols
NBAR VRF-Aware	Cisco IOS XE Release 3.3S	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following section provides information about this feature: NBAR Scalability
NBAR Multi stage Classification	Cisco IOS XE Release 3.7S	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following section provides information about this feature: NBAR Multi-stage Classification.
NBAR2: Add/Rename Static Attributes	Cisco IOS XE Release 3.11S	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**.

Feature Name	Releases	Feature Information
NBAR2 GETVPN (Cryptomap) Support	Cisco IOS XE Release 3.11S	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following section provides information about this feature: NBAR Support for GETVPN, on page 18
NBAR Support for CAPWAP	Cisco IOS XE Release 3.17S	CAPWAP (Control And Provisioning of Wireless Access Points) is a protocol is used in wireless traffic, providing point-to-point encapsulation (tunnel) for application traffic. NBAR provides a CAPWAP recognition mode that enables NBAR classification of the application traffic within a CAPWAP tunnel. The following section provides information about this feature: NBAR Support for CAPWAP
NBAR DNS-based Classification	Cisco IOS XE Release 3.17S	This feature can improve traffic classification by using DNS transaction information exchanged when a user initiates a connection with an application server. This method offers the significant advantage of classifying flows from the first packet in the flow. The following section provides information about this feature: NBAR DNS-based Classification
Customizing Built-in Protocols	Cisco IOS XE Denali 16.3	Customizing an NBAR built-in protocol (provided by the Cisco Protocol Pack) to include an additional user-specified domain extends the scope of the built-in protocol. Any policy associated with the protocol will then apply to the user-specified domain also. The following section provides information about this feature: Customizing Built-in Protocols

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.

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 the policy maps to interfaces. Policy maps are used to apply the appropriate quality of service (QoS) to network traffic.

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.

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.

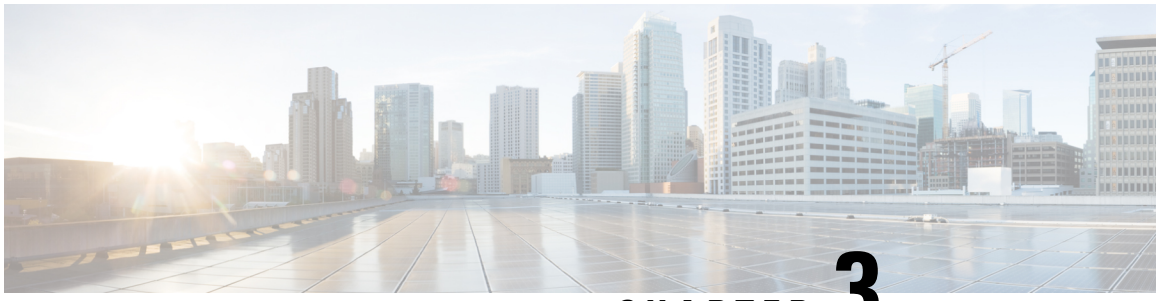
Subport classification—The classification of network traffic by information that is contained in the packet payload, that is, information found beyond the TCP or UDP port number.

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 3

NBAR2 Protocol Pack

The NBAR2 Protocol Pack provides an easy way to update protocols supported by NBAR2 without replacing the base IOS image that is already present in the device. A Protocol Pack is a set of protocols developed and packaged together. To view the list of protocols supported in a Protocol Pack, see NBAR2 Protocol Library.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for the NBAR2 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 NBAR2 Protocol Pack that is the exact match for the NBAR2 engine, and also load the latest rebuild of Cisco software. See the NBAR2 Protocol Library page for compatibility information.

Information About the NBAR Protocol Pack

Protocol Pack Overview

NBAR2 Protocol Packs are software packages that update the protocol support on a device without replacing the Cisco software on the device. A Protocol Pack contains a set of signatures supported by NBAR2.

Protocol Packs are sets of protocols developed and packaged together. Each Cisco IOS image comes with a built-in Protocol Pack. With a standard license, a subset of protocols and Protocol Pack features are supported. With an advanced license, all protocols and features are supported. Updating the Protocol Pack on a Cisco IOS release requires an advanced license. For information about licensing, see AVC Licensing and Feature Activation.

To view the list of protocols supported in a Protocol Pack, see NBAR2 Protocol Library.

The NBAR2 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 NBAR2 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*.

Protocols Available with Standard License

The default Protocol Pack available with a standard license includes the protocols shown below. For information about the Protocol Packs available with an advanced license, see the NBAR2 Protocol Library.

bgp
bittorrent
cifs
citrix
cuseeme
dhcp
dht
directconnect
dns
edonkey
egp
eigrp
exchange
fasttrack
finger
ftp
gnutella
gopher

gre
http
http-local-net
https
icmp
imap
ipinip
ipsec
ipv6-icmp
irc
kazaa2
kerberos
l2tp
ldap
mgcp
ms-rpc
netbios
nfs
nntp
notes
novadigm
ntp
ospf
pop3
pptp
printer
rip
rsvp
rtcp
rtp
rtsp
secure-ftp
secure-http
secure-imap
secure-irc
secure-ldap
secure-nntp
secure-pop3
secure-telnet
sip
skinny
skype
smtp
snmp
socks
sqlnet

sqlserver
ssh
ssl
stun-nat
sunrpc
syslog
telepresence-control
telnet
teredo-ipv6-tunneled
tftp
winmx
xmpp-client
xwindows

SSL Unique-name Sub-classification

The "unique-name" sub-classification parameter can be used to match SSL sessions of servers that are not known globally, or are not yet supported by NBAR2. The unique-name will match the server name indication (SNI) field in the client request if the SNI field exists, or it will match the common name (CN) field in the first certificate of the server's response.

Note The SSL sub-classification parameters have priority over the built in signatures. Therefore, when a unique-name defined by a user matches a known application such as Facebook, it will not match the built-in protocol but will match SSL with the configured sub-classification.

Note Similar to the other sub-classification features, the classification result (for example, as seen in protocol-discovery), does not change and will remain as SSL. However, the flows matching the class maps will receive the services such as QoS and Performance monitor configured for them. To view the detailed matching statistics, refer to the policy map counters.

For more information on SSL, see http://tools.ietf.org/html/rfc6101.

RTP Dynamic Payload Type Sub-classification

The sub-classification parameters for Real-time Transport Protocol (RTP) audio and RTP video detect RTP flows that use dynamic payload types (PT). Dynamic PTs are PTs in the dynamic range from 96 to 127, as defined in the RTP RFC, and are used by protocols such as SIP and RTSP.

Note The RTP audio/video sub-classification parameters are generic in nature and will match only on generic RTP traffic. More specific classification such as ms-lync-audio, cisco-jabber-audio, facetime, and cisco-phone will not match as RTP, and therefore will not match the audio/video sub-classification.

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

Example: Loading the NBAR2 Protocol Pack

The following example shows how to load an NBAR2 Protocol Pack named defProtoPack from the harddisk:

```
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 NBAR2 Protocol Pack:

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

The following example shows how to load a Protocol Pack of a lower version using the **force** keyword:

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

Example: Verifying the Loaded NBAR2 Protocol Pack

The following sample output from the **show ip nbar protocol-pack active** command shows information about the Protocol Pack that is provided by default 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.
NBAR Engine Version:            14
```

The following sample output from the **show ip nbar protocol-pack active detail** command shows detailed information about the active Protocol Pack that is provided by default 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.
NBAR Engine Version:             14
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 an advanced Protocol Pack that is present in the specified device location:

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

Name:                            Advanced Protocol Pack
Version:                         2.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             14
Creation time:                   Mon Jul 16 09:29:34 UTC 2012
```

The following sample output from the **show ip nbar protocol-pack** command shows detailed protocol pack information present in the specified disk location:

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

Name:                            Advanced Protocol Pack
Version:                         2.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             14
Creation time:                   Mon Jul 16 09:29:34 UTC 2012
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>
  .
  .
  .
  .
```

Example: Classifying SSL Sessions

The following example shows how an SSL-based service with the server name as 'finance.cisco.com' is matched using **unique-name**:

```
Device> enable
Device# configure terminal
Device(config)# class-map match-any cisco-finance
Device(config-cmap)# match protocol ssl unique-name finance.cisco.com
```

Example: Classifying RTP Dynamic Payload Type

The following example shows how to detect RTP audio flows that include both static and dynamic PT:

```
Device> enable
Device# configure terminal
Device(config)# class-map match-any generic-rtp-audio
Device(config)# match protocol rtp audio
```

Additional References for NBAR2 Protocol Pack

Related Documents

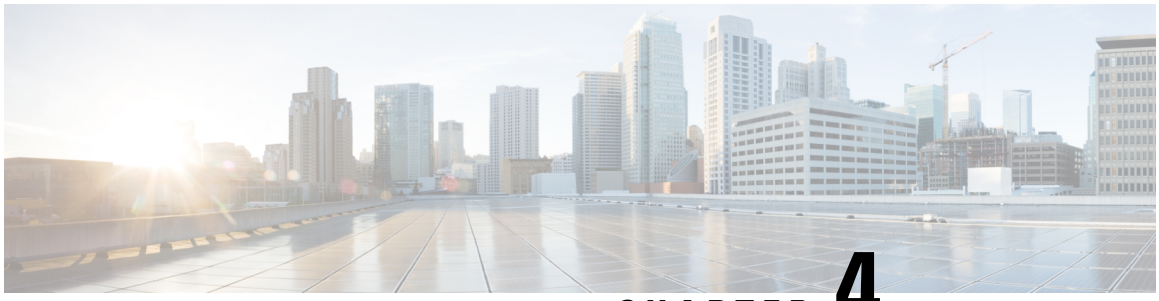
Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS LAN Switching commands	Cisco IOS LAN Switching Command Reference
Cisco IOS QoS configuration information	QoS Configuration Guide

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



CHAPTER 4

Enabling Protocol Discovery

Network-Based Application Recognition (NBAR) includes a feature called Protocol Discovery. Protocol discovery provides an easy way to discover the application protocol packets that are passing through 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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Enabling Protocol Discovery

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

Restrictions for Enabling Protocol Discovery

NBAR protocol discovery does not support the following:

- Asymmetric flows with stateful protocols.

Note In the NBAR context, asymmetric flows are the flows in which different packets of the flow go through different routers, for reasons such as load balancing implementation or asymmetric routing where packets flow through different routes to different directions.

- NBAR processing. By design, NBAR processing is temporarily disabled during the In-Service Software Upgrade (ISSU). The following syslog message indicates restart of NBAR classification once ISSU is complete.

"%NBAR_HA-5-NBAR_INFO: NBAR sync DONE!"

- Multicast packet classification.
- 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.
- Non-IP traffic.
- Packets that originate from or that are destined to the router running NBAR.

NBAR is not supported on the following logical interfaces:

- Dialer interfaces
- Dynamic tunnels such as Dynamic Virtual Tunnel Interface (DVTI)
- IPv6 tunnels that terminate on the device
- MPLS
- Overlay Transport Virtualization (OTV) overlay interfaces

Note In cases where encapsulation is not supported by NBAR on some links, you can apply NBAR on other interfaces of the device to perform input classification. For example, you can configure NBAR on LAN interfaces to classify output traffic on the WAN link.

The following virtual interfaces are supported depending on the image of your Cisco IOS:

- Generic routing encapsulation (GRE)
- IPsec IPv4 tunnel (including tunneled IPv6) in protocol discovery mode and MQC mode
- IPsec IPv6 tunnel in protocol discovery mode but not in MQC mode
- Multipoint GRE/Dynamic Multipoint VPN (DMVPN) in protocol discovery mode

Note NBAR requires more CPU power when NBAR is enabled on tunneled interfaces.

If protocol discovery is enabled on both the tunnel interface and the physical interface on which the tunnel interface is configured, the packets that are designated to the tunnel interface are counted on both interfaces. On the physical interface, the packets are classified and are counted based on the encapsulation. On the tunnel interface, packets are classified and are counted based on the Layer 7 protocol.

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.

Information About Protocol Discovery

Protocol Discovery Overview

The Protocol Discovery feature of NBAR provides an easy way of discovering the application protocols passing through an interface so that appropriate QoS features can be applied.

NBAR determines which protocols and applications are currently running on your network. Protocol discovery provides an easy way of discovering the application protocols that are operating on an interface so that appropriate 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

These statistics can be used when you 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.

Interface Scalability

Depending on your release, there is a limit on the number of interfaces on which protocol discovery can be enabled.

The following table provides the details of the protocol discovery supported interface and the release number:

Table 7: Release and Protocol Discovery Interface Support

Release	Number of Interfaces Supported with Protocol Discovery
Releases prior to Cisco IOS XE Release 2.5	No restriction
Cisco IOS XE Release 2.5	128
Cisco IOS XE Release 2.6	256
Cisco IOS XE Release 2.7	32
Cisco IOS XE Release 3.2S and later	32

How to Enable Protocol Discovery

Enabling Protocol Discovery on an Interface

Perform this task to enable protocol discovery on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **ip nbar protocol-discovery** [**ipv4** | **ipv6**]
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 fastethernet1/1/1`	Configures an interface type and enters interface configuration mode. • Enter the interface type and the interface number.

	Command or Action	Purpose
Step 4	**ip nbar protocol-discovery** [**ipv4** | **ipv6**] **Example:** `Router(config-if)# ip nbar protocol-discovery`	Configures NBAR to discover traffic for all protocols that are known to NBAR on a particular interface. • (Optional) Enter the **ipv4** keyword to enable protocol discovery statistics collection for IPv4 packets, or enter the **ipv6** keyword to enable protocol discovery statistics collection for IPv6 packets. • Specifying either of these keywords enables the protocol discovery statistics collection for the specified IP version only. If neither keywords is specified, statistics collection is enabled for both IPv4 and IPv6. • The **no** form of this command is not required to disable a keyword because the statistics collection is enabled for the specified keyword only.
Step 5	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode.

Reporting Protocol Discovery Statistics

Perform this task to display a report of the protocol discovery statistics per interface.

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 1/1/1`	(Optional) Displays the packet and class statistics for all policy maps on the specified interface. • Enter the interface type and interface number.

	Command or Action	Purpose
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 Fastethernet1/1/1`	Displays the statistics gathered by the NBAR Protocol Discovery feature. • (Optional) Enter keywords and arguments to fine-tune the statistics displayed. For more information on each of the keywords, refer to the **show ip nbar protocol-discovery** command in Cisco IOS Quality of Service Solutions Command Reference.
Step 4	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuration Examples for Protocol Discovery

Example: Enabling Protocol Discovery on an Interface

In the following sample configuration, protocol discovery is enabled on Fast Ethernet interface 1/1/1:

```
Router> enable
Router# configure terminal
Router(config)# interface fastethernet1/1/1
Router(config-if)# ip nbar protocol-discovery
Router(config-if)# end
```

In the following sample configuration, protocol discovery is enabled on Fast Ethernet interface 1/1/2 for IPv6 packets:

```
Router> enable
Router# configure terminal
Router(config)# interface fastethernet1/1/2
Router(config-if)# ip nbar protocol-discovery ipv6
Router(config-if)# end
```

In the following sample configuration, protocol discovery is enabled on Fast Ethernet interface 1/1/2 for IPv6 packets. Later, the protocol discovery is enabled for IPv4 packets and this does not require the **no** form for the **ipv6** keyword.

```
Router> enable

Router# configure terminal

Router(config)# interface fastethernet1/1/2

Router(config-if)# ip nbar protocol-discovery ipv6

Router(config-if)# ip nbar protocol-discovery ipv4

Router(config-if)# end
```

Example: Reporting Protocol Discovery Statistics

The following sample output from the **show ip nbar protocol-discovery** command displays the five most active protocols on the Fast Ethernet interface 2/0/1:

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

 FastEthernet2/0/1
                           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
```

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*
Concepts and information about NBAR	"Classifying Network Traffic Using NBAR" module
MQC	"Applying QoS Features Using the MQC" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

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

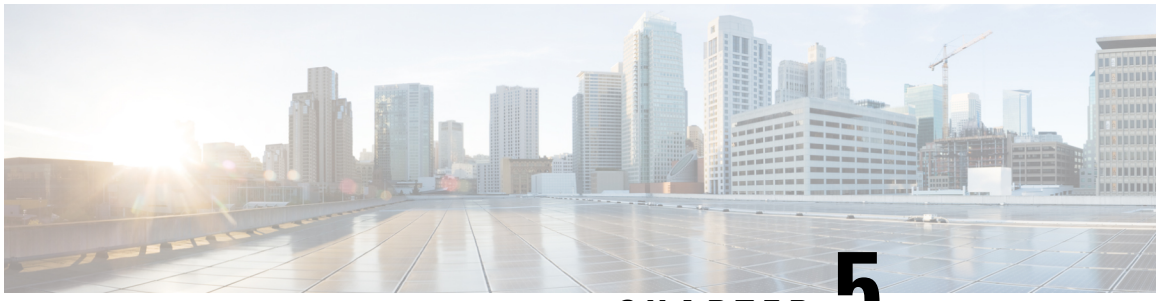
Feature Information for Enabling Protocol Discovery

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 8: Feature Information for Enabling Protocol Discovery

Feature Name	Releases	Feature Information
Protocol Discovery	Cisco IOS XE 2.1 Cisco IOS XE 3.3S	This feature was introduced on Cisco ASR 1000 Series Routers. The following sections provide information about this feature: The following commands were introduced: **ip nbar protocol discovery, show ip nbar protocol discovery.**



CHAPTER 5

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 www.cisco.com/go/cfn. 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 60.

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 9: 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

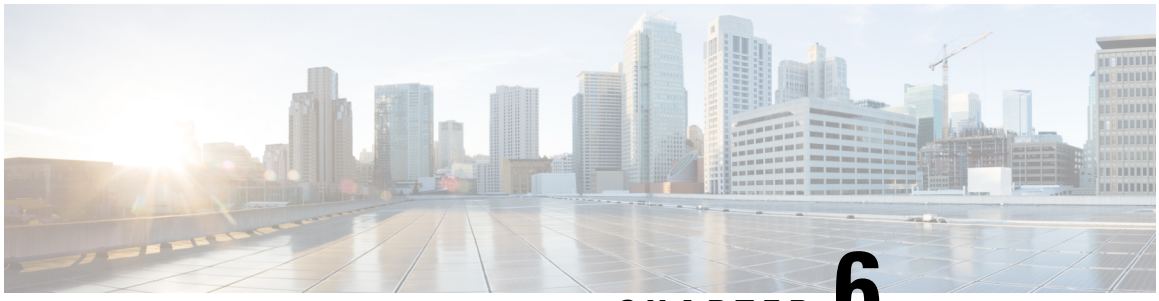
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 10: 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 6

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 www.cisco.com/go/cfn. 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

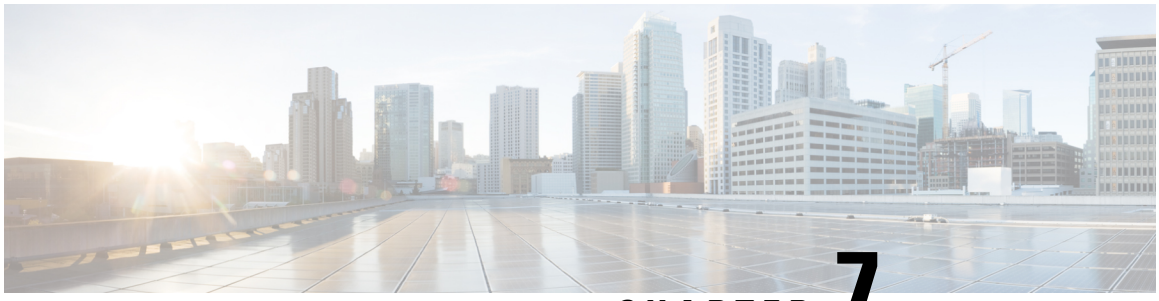
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 11:

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 7

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 www.cisco.com/go/cfn. 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

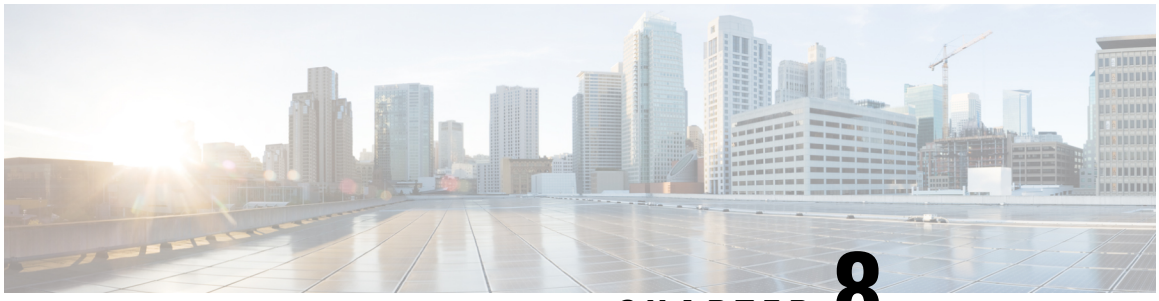
Feature Information for MQC Based on Transport Hierarchy

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 12: Feature Information for MQC Based on Transport Hierarchy

Feature Name	Releases	Feature Information
MQC Based on Transport Hierarchy	Cisco IOS XE 3.14S	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 8

NBAR Categorization and Attributes

NBAR Categorization and Attributes feature provides the mechanism to match protocols or applications based on statically assigned attributes such as application-group, category, sub-category, encrypted and tunnel. Categorizing the protocols and applications into different groups helps with reporting and applying Quality of Service (QoS) policies.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About NBAR2 Custom Protocol

NBAR Categorization and Attributes

The NBAR Categorization and Attributes feature provides the mechanism to match protocols or applications based on certain attributes. Categorizing the protocols and applications into different groups will help with reporting and 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:

- **application-group**: The **application-group** keyword 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** keyword 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** keyword 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** keyword 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 the encrypted and nonencrypted status of the applications. Protocols for which the NBAR does not provide any value are categorized under the unassigned encrypted group.
- **tunnel**: The **tunnel** keyword 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 configurations do 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.

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.

How to Configure NBAR2 Custom Protocol

Customizing NBAR Attributes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar attribute-map** *profile-name*
4. [**attribute category** *category-name*]
5. [**attribute sub-category** *sub-category-name*]
6. [**attribute application-group** *application-group-name*]
7. [**attribute tunnel** *tunnel-info*]
8. [**attribute encrypted** *encrypted-info*]
9. [**attribute traffic-class** *traffic-class*]
10. [**attribute business-relevance** *business-relevance*]
11. [**attribute p2p-technology** *p2p-technology-info*]
12. **ip nbar attribute-set** *protocol-name profile-name*
13. **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 attribute-map** *profile-name* **Example:** `Device(config)# ip nbar attribute-map actdir-attrib`	Creates an attribute profile with the name that you specify, and enters the attribute-map configuration mode.
Step 4	[**attribute category** *category-name*] **Example:** `Device(config-attribute-map)# attribute category net-admin`	Adds attribute values from the application-group attribute, on to your profile.

	Command or Action	Purpose
Step 5	[**attribute sub-category** *sub-category-name*] **Example:** `Device(config-attribute-map)# attribute sub-category network-management`	Adds attribute values from the sub-category attribute, on to your profile.
Step 6	[**attribute application-group** *application-group-name*] **Example:** `Device(config-attribute-map)# attribute application-group other`	Adds attribute values from the application-group attribute, on to your profile.
Step 7	[**attribute tunnel** *tunnel-info*] **Example:** `Device(config-attribute-map)# attribute tunnel no`	Adds attribute values from the tunnel attribute, on to your profile.
Step 8	[**attribute encrypted** *encrypted-info*] **Example:** `Device(config-attribute-map)# attribute encrypted no`	Adds attribute values from the encrypted attribute, on to your profile.
Step 9	[**attribute traffic-class** *traffic-class*] **Example:** `Device(config-attribute-map)# attribute traffic-class multimedia-conferencing`	Adds traffic-class attribute value to the profile.
Step 10	[**attribute business-relevance** *business-relevance*] **Example:** `Device(config-attribute-map)# attribute business-relevance business-relevant`	Adds business-relevance attribute value to the profile.
Step 11	[**attribute p2p-technology** *p2p-technology-info*] **Example:** `Device(config-attribute-map)# attribute p2p-technology no`	Adds attribute values from the p2p-technology attribute, on to your profile.
Step 12	**ip nbar attribute-set** *protocol-name profile-name* **Example:** `Device(config-attribute-map)# ip nbar attribute-set active-directory actdir-attrib`	Adds attribute values from the specified profile to the specified protocol.

	Command or Action	Purpose
Step 13	**end** **Example:** `Device(config-attribute-map)# end`	Returns to privileged EXEC mode.

Configuration Examples for NBAR2 Custom Protocol

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
```

Example: Creating a Profile and Configuring Attributes for the Profile

The following example shows how to create an attribute profile with attributes configured for the Network News Transfer Protocol (NNTP) protocol:

```
Device> enable
Device# configure terminal
Device(config)# ip nbar attribute-map nntp-attrib
Device(config-attribute-map)# attribute category newsgroup
Device(config-attribute-map)# attribute application-group nntp-group
Device(config-attribute-map)# attribute tunnel tunnel-no
Device(config-attribute-map)# attribute encrypted encrypted-yes
Device(config-attribute-map)# attribute p2p-technology p2p-tech-no
Device(config-attribute-map)# end
```

The following example shows how to verify the above configuration:

```
Device> enable
Device# show ip nbar attribute-map nntp-attrib
Device# Profile Name :  nntp-attrib
            category :  newsgroup
   application-group :  nntp-group
           encrypted :  encrypted-yes
Device# end
```

Example: Attaching an Attribute Profile to a Protocol

The following example shows how to set an attribute profile to the Application Communication Protocol (ACP) protocol:

```
Device> enable
Device# configure terminal
```

```
Device(config)# ip nbar attribute-set acp test-profile
Device(config)# exit
```

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 NBAR Categorization and Attributes

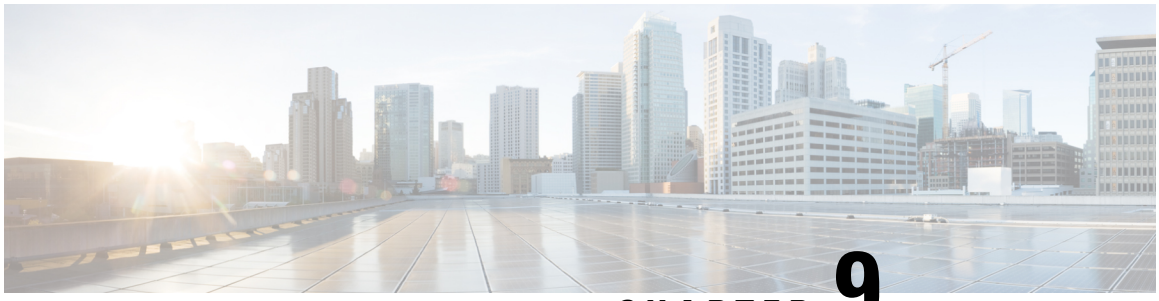
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train 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 NBAR2 Custom Protocol

Feature Name	Releases	Feature Information
NBAR Categorization and Attributes	Cisco IOS XE Release 3.4S	This feature was introduced on Cisco ASR 1000 series Aggregation Services Routers. The following command was introduced or modified: **ip nbar custom**

Feature Name	Releases	Feature Information
NBAR2 Custom Protocol	Cisco IOS XE Release 3.8S	The NBAR2 Custom Protocol feature configures attributes profiles for protocols, and maps profiles to protocols. The following command was introduced or modified: **ip nbar attribute-map**, **ip nbar attribute-set**.



CHAPTER 9

Reporting Extracted Fields Through Flexible NetFlow

The Reporting Extracted Fields Through Flexible NetFlow feature allows Network-Based Application Recognition (NBAR) to send subapplication table fields to the collector through Flexible NetFlow.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Reporting Extracted Fields Through Flexible NetFlow

Subapplication Table Fields

Use the **option sub-application-table** command to send an options table periodically to the collector, thereby enabling the collector to map NBAR subapplication tags, subapplication names, and subapplication descriptions provided in the flow records to application IDs.

How to Report Extracted Fields Through Flexible NetFlow

Reporting Subapplication Table Fields

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **option sub-application-table**
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	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Enters Flexible NetFlow flow exporter configuration mode.
Step 4	**option sub-application-table** **Example:** `Device(config-flow-exporter)# option sub-application-table`	Enables periodic sending of an options table that allows the collector to map NBAR subapplication tags, subapplication names, and subapplication descriptions provided in flow records to application IDs.
Step 5	**exit** **Example:** `Device(config-flow-exporter)# exit`	Exits Flexible NetFlow flow exporter configuration mode and returns to global configuration mode.

Configuration Examples for Reporting Extracted Fields Through Flexible NetFlow

Example: Reporting Subapplication Fields

The following example shows how to enable the periodic sending of an options table, which allows the collector to map NBAR subapplication tags, subapplication names, and subapplication descriptions provided in the flow records to application IDs:

```
Device(config)# flow exporter FLOW-EXPORTER-1
Device(config-flow-exporter)# option sub-application-table
```

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

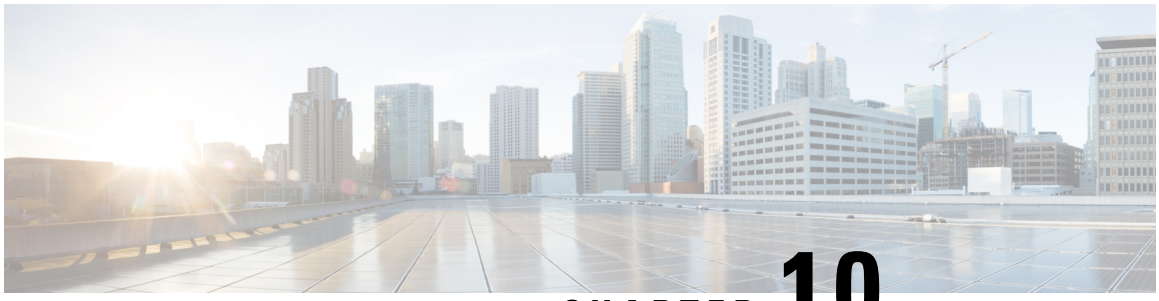
Feature Information for Reporting Extracted Fields Through Flexible NetFlow

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 Reporting Extracted Fields Through Flexible NetFlow

Feature Name	Releases	Feature Information
Reporting Extracted Fields Through Flexible NetFlow	Cisco IOS XE Release 3.7	The Reporting Extracted Fields Through Flexible NetFlow feature allows NBAR to send subapplication table fields to the collector through Flexible NetFlow. The following command was introduced or modified: **option (Flexible NetFlow)**.



CHAPTER 10

NBAR Protocol Pack Auto Update

Important Beginning with Cisco IOS XE Fuji 16.8.1, this feature has been deprecated. Use Cisco Software-Defined AVC (SD-AVC) for automating Protocol Pack deployment. SD-AVC, a component of Cisco Application Visibility and Control (AVC), uses a centralized network service that operates with devices in a network to provide numerous services, including Protocol Pack deployment.

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 100
Maintenance	Keeping Protocol Packs Up-to-Date, on page 106

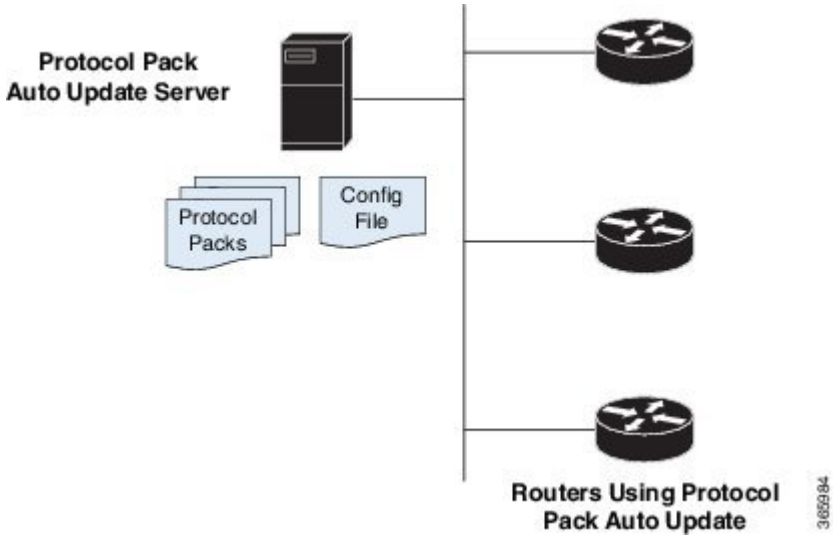
Topic	Section
Router Procedures	Enabling Protocol Pack Auto Update, on page 107 Disabling Protocol Pack Auto Update, on page 108 Initiating Immediate Protocol Pack Update, on page 109 Displaying Protocol Pack Auto Update Information, on page 109

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 2: 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 107.

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 101.

2. On participating routers, enable Protocol Pack Auto Update.

   See Enabling Protocol Pack Auto Update, on page 107.

   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 110.

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 102 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.

Parameter	Description
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
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:**	Exits global configuration mode.

	Command or Action	Purpose
	`Device(config)# exit`	

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.
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 111.
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 110.

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

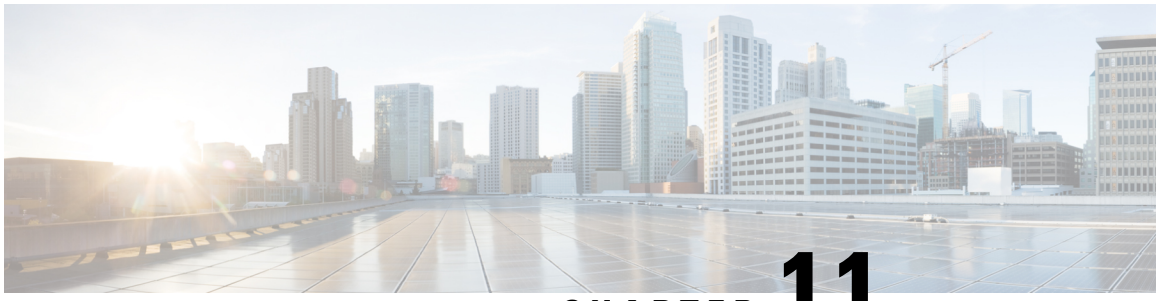
Command	Description
clear-previous {**enable** | **disable**}	**enable**: Causes unneeded Protocol Pack files to be removed after a cool-down period. **disable**: Configures the feature to not remove any files. Default: Enable
force-upgrade {**enable** | **disable**}	**enable**: New Protocol Pack updates will be applied with the "force" flag. **disable**: New Protocol Pack updates will not be applied with the "force" flag. Default: Disable

Command	Description
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 11

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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.

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.

IP address-based custom application gets classification from syn packet.

Comparison of Custom NBAR Protocols: Based on a Single Network Protocol or 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).

NBAR provides:

- **Custom NBAR protocols based on single network protocol**

  Useful for identifying a single type of traffic (HTTP, SSL, and so on) according to a specified pattern.

  Syntax: ip nbar custom <protocol_name> <traffic_type> <criteria>

- **Custom NBAR protocols based on multiple network protocols** (called a "composite" custom NBAR protocol)

  Useful for identifying traffic using signatures for multiple network protocols. Currently, the composite method provides an option, "server-name" (value for <composite_option> in the CLI syntax) that 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.

  Syntax: ip nbar custom <protocol_name> composite <composite_option> <criteria>

Example Use Case: Custom NBAR Protocol Based on Multiple Network Protocols

- **Objective**: Identify all HTTP, SSL, and DNS traffic associated with the abc_example.com server.
- **Preferred method**: Use a composite custom NBAR protocol.

- **CLI**: `ip nbar custom abc_example_custom composite server-name *abc_example`

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:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Router> enable`	
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*] **Example:** `Router(config)# ip nbar custom app_sales1 5 ascii SALES source tcp 4567`	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 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.

	Command or Action	Purpose
		• 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.

	Command or Action	Purpose
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.)
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.

	Command or Action	Purpose
		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.

	Command or Action	Purpose
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** **Example:** `Router(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:** `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.
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.

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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• 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.
	`Device(config)# ip nbar custom mycustomprotocol transport tcp`
	`Device(config-custom)# ip subnet 255.255.255.255 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.

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 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.

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

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

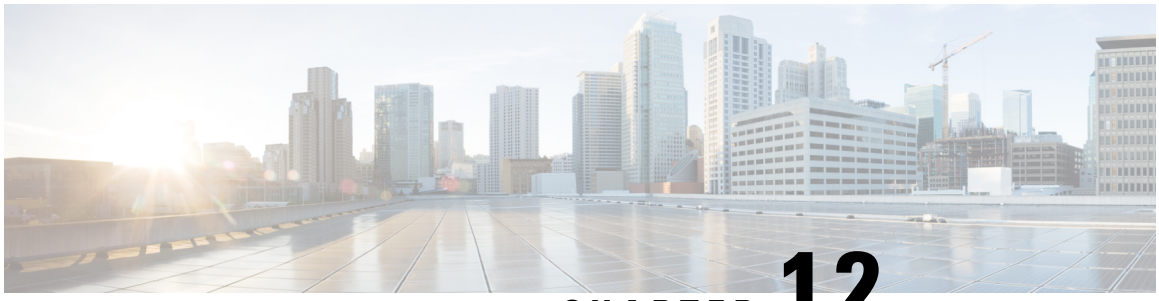
Feature Information for NBAR2 Custom Protocol

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 15: Feature Information for NBAR2 Custom Protocol

Feature Name	Releases	Feature Information
NBAR2 Custom Protocol	Cisco IOS XE Release 3.8S	This feature was introduced on Cisco ASR 1000 series Aggregation Services Routers. The following command was introduced or modified: **ip nbar custom**
NBAR2 Custom Protocol Enhancements Ph II	Cisco IOS XE Release 3.12S	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 12

NBAR2 Protocol Pack Hitless Upgrade

The NBAR2 Protocol Pack Hitless Upgrade feature enables users to seamlessly upgrade a Network-Based Application Recognition (NBAR) protocol pack or change the NBAR configurations without impacting any of the current classification configurations on a device.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for NBAR2 Protocol Pack Hitless Upgrade

Additional memory is required to support the NBAR2 Protocol Pack Hitless Upgrade feature because it holds together two configurations until the previous configuration is aged.

Information About NBAR2 Protocol Pack Hitless Upgrade

Overview of NBAR2 PP Hitless Upgrade

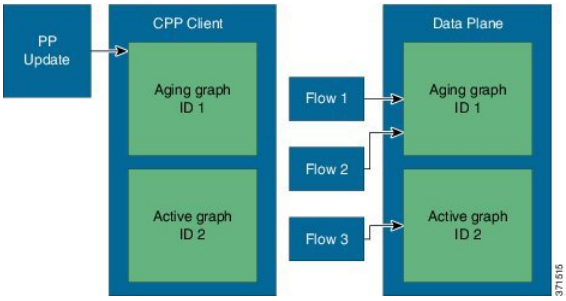
Hitless Upgrade is the method to upgrade the NBAR2 Protocol Pack (PP) components on an NBAR engine without incurring any service downtime. In earlier Cisco IOS software releases, NBAR could hold only a single configuration graph on the control plane client that is transferred to the data path. From Cisco IOS XE Release 3.12S onward, NBAR can hold several configurations graphs at a single time. When a new configuration

change occurs, a new configuration graph is created on the control plane client. The new graph is downloaded to the data plane, and all new flows are directed to the new graph.

If a packet arrives from a flow that was being classified, the packet is directed to the correct configuration graph (the one that was active when the flow was created).

The following illustration displays the NBAR system state after a configuration or protocol pack update:

Figure 3: Aging a Graph



In the illustration above, when a new graph is created, the old graph is moved to the aging state. In an aged state, only flows that are associated with the graph are referenced with the graph. If a flow is not classified until aging time, it is reported as unknown by NBAR.

Note Due to memory limitations, it is important to limit the number of parallel existing graphs and aging graphs in the NBAR system. Currently, all platforms can hold a maximum two configurations at a given time.

Use the **show platform software nbar statistics** command to view the status of NBAR.

Benefits of NBAR2 Protocol Pack Hitless Upgrade

NBAR2 Protocol Pack Hitless Upgrade provides the following benefits:

- No loss of information for classified flows during a protocol upgrade
- No impact on new flows
- No impact on in-progress flows

Additional References for NBAR2 Protocol Pack Hitless Upgrade

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NBAR Protocol Pack	*QoS: NBAR Configuration Guide*

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

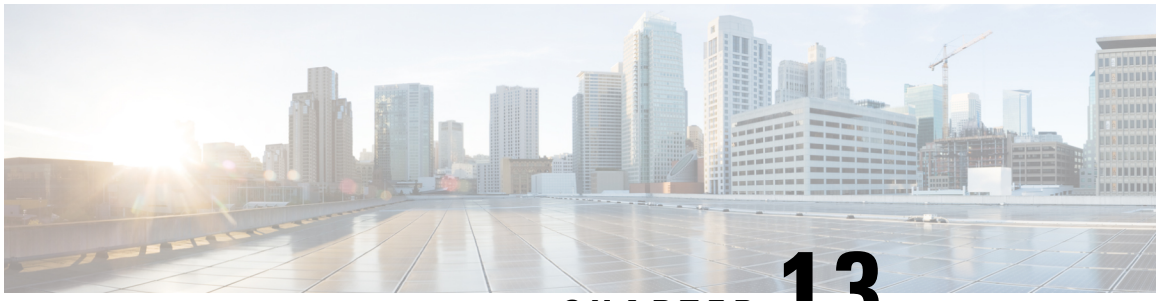
Feature Information for NBAR2 Protocol Pack Hitless Upgrade

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 Information for NBAR2 Protocol Pack Hitless Upgrade

Feature Name	Releases	Feature Information
NBAR2 Protocol Pack Hitless Upgrade	Cisco IOS XE Release 3.12S	The NBAR2 Protocol Pack Hitless Upgrade feature enables seamless upgrade of a NBAR protocol pack or NBAR configurations without impacting any of the current classification configurations on a device. In Cisco IOS XE Release 3.12S, support was added for the Cisco ASR 1000 Series Routers.



CHAPTER 13

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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:**	Defines web-based custom protocol match. • Enter the custom protocol name and any other optional keywords and arguments.

	Command or Action	Purpose
	`Router(config)# ip nbar custom app_sales1 http url www.example.com`	**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

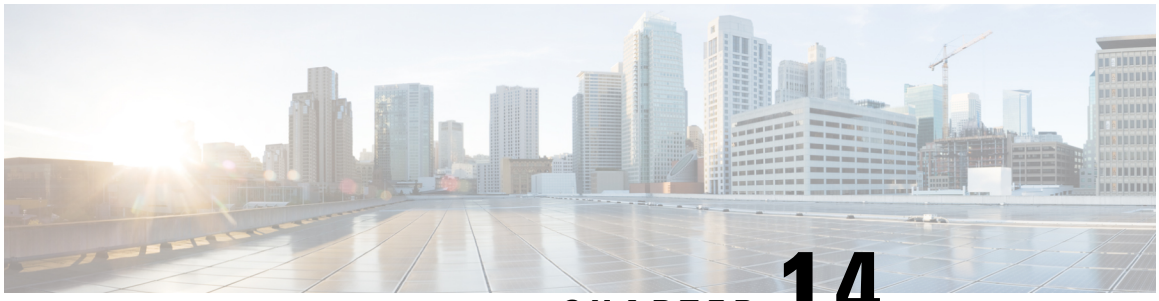
Feature Information for NBAR Web-based Custom Protocols

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 17: Feature Information for NBAR Web-based Custom Protocols

Feature Name	Releases	Feature Information
NBAR Web-based Custom Protocols Scalability	Cisco IOS XE Release 3.13S	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 14

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 www.cisco.com/go/cfn. 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 141, 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 142.

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:

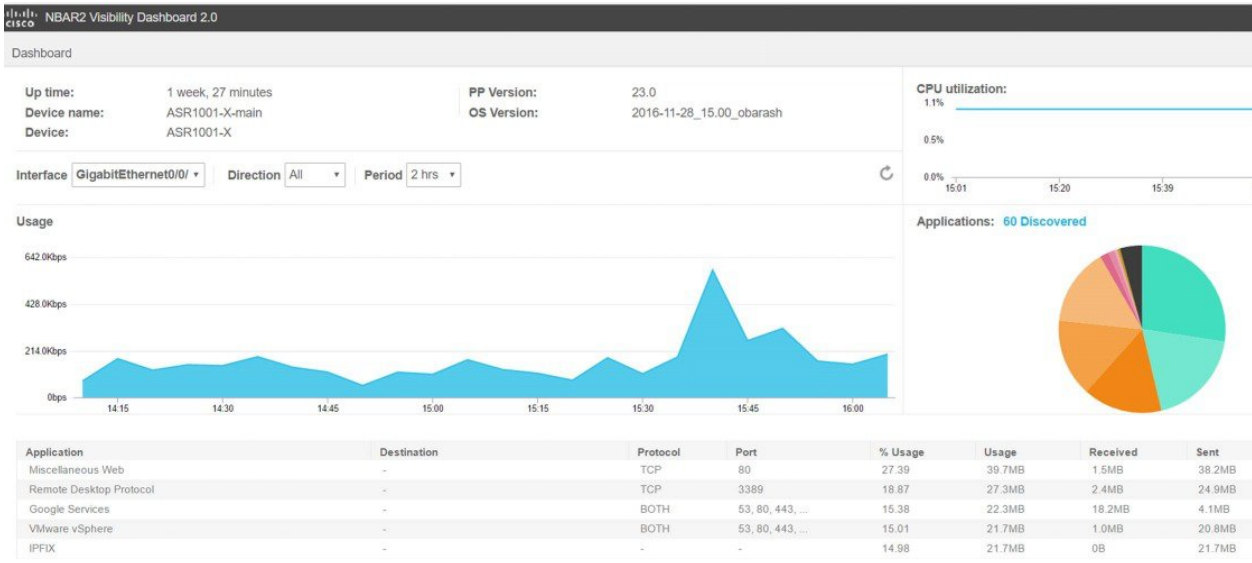
     ```
     Device(config)#ip http authentication enable
     Device(config)#ip http authentication local
     Device(config)#username cisco
     Device(config)#password n449rbpsvq
     ```

   - 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 4: Visibility Dashboard



Configuring NBAR2 HTTP-Based Visibility Dashboard

Before you begin

The HTTP-based Visibility Dashboard uses the Protocol Discovery feature. For details about Protocol Discovery, see How to Enable Protocol Discovery, on page 52.

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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device#configure terminal`	
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`	Enables protocol discovery. For more information, see How to Enable Protocol Discovery, on page 52.

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

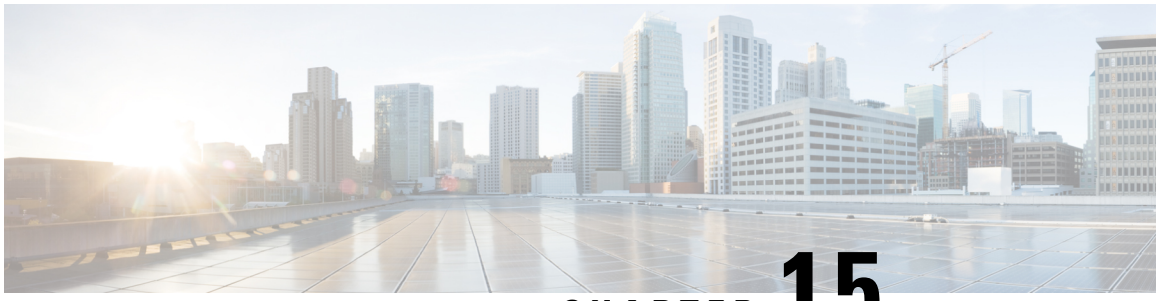
Feature Information for NBAR2 HTTP-Based Visibility Dashboard

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 18: 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 15

NBAR Coarse-Grain Classification

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

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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

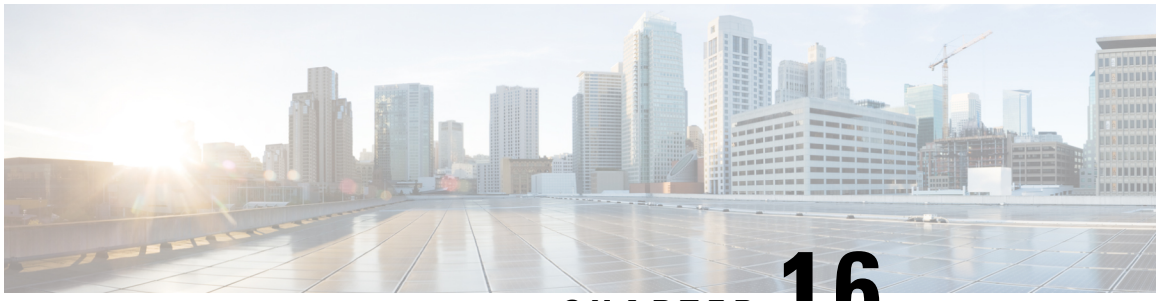
Feature Information for NBAR Coarse-Grain Classification

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 19: Feature Information for NBAR Coarse-Grain Classification

Feature Name	Releases	Feature Information
NBAR Coarse-Grain Classification	Cisco IOS XE Release 3.14S	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 16

SSL Custom Application

SSL Custom Application feature enables users to customize applications that run on any protocol over Secure Socket Layer (SSL), including HTTP over Secure Socket Layer (HTTPS), using the server name, if it exists in the Client Hello extensions, or the common name from the certificate that the server sends to the client.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About SSL Custom Application

Overview of SSL Custom Application

SSL Custom Application feature enables users to customize applications that run on any protocol over Secure Socket Layer (SSL), including HTTP over Secure Socket Layer (HTTPS), using the server name, if it exists in the Client Hello extensions, or the common name from the certificate that the server sends to the client.

HTTP over Secure Socket Layer (HTTPS) is a communication protocol for secure communication. HTTPS is the result of layering HTTP on SSL protocol.

In SSL sub-classification, the rule that ends later in the packet will match. For example, consider the server name ‘finance.example.com’, if there is a rule for ‘finance’ and another rule for example.com, then the rule for ‘example.com’ will match.

SSL Unique Name Sub-Classification

The SSL unique-name parameter is used to match SSL sessions of servers that are not known globally, or are not yet supported by NBAR. The unique-name matches the server name indication (SNI) field in the client request, if the SNI field exists, or it matches the common name (CN) field in the first certificate of the server's response.

The feature also supports cases of SSL sessions that use session-id than the SSL sessions that use handshake.

The server name is available as part of a HTTPS URL itself. For example, in the URL https://www.facebook.com, the server name is www.facebook.com. However, the certificate is found in the browser. The user can observe the certificate information by clicking on the HTTPS icon.

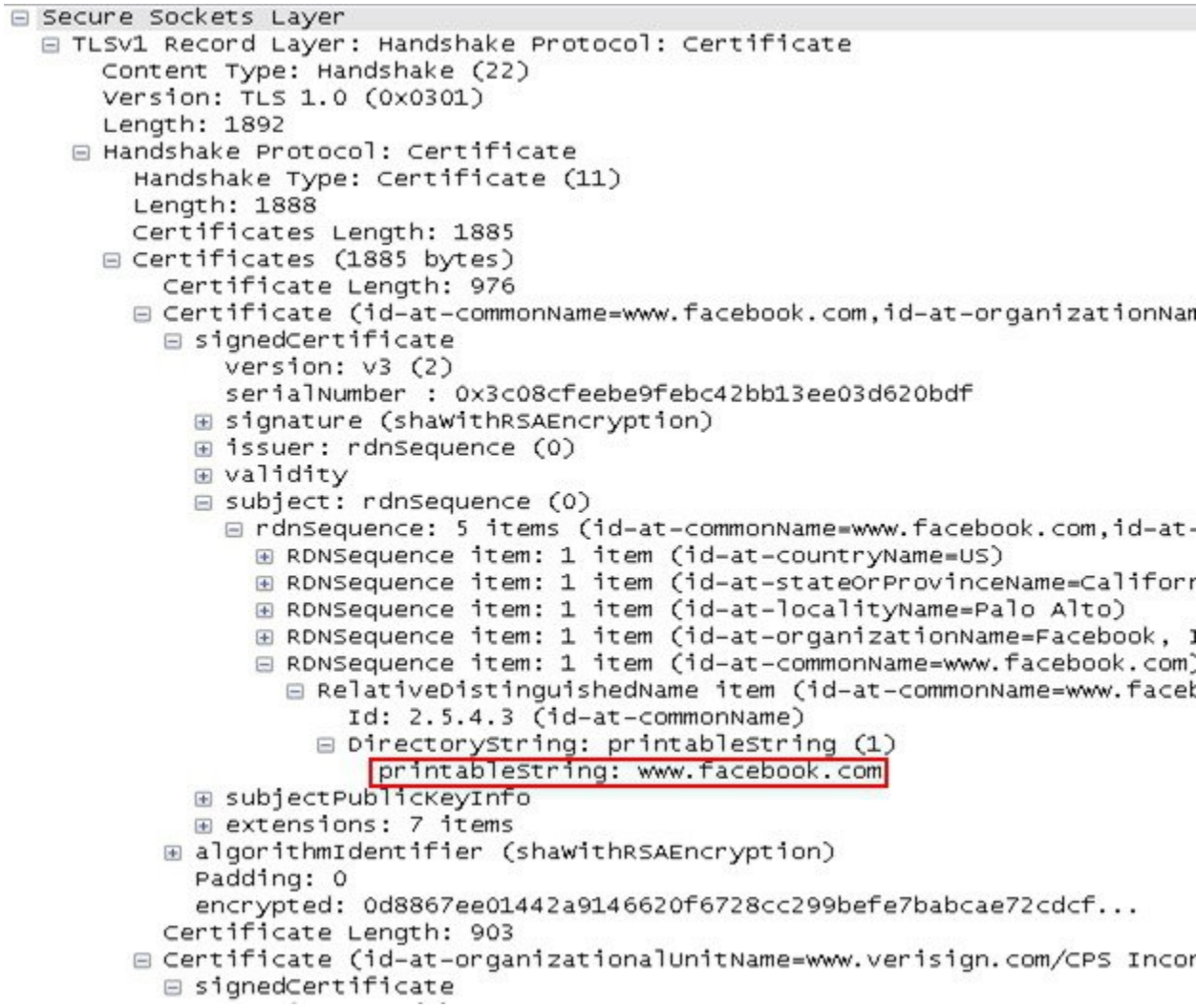
The following two figures display the location of the server name and common name as it is visible to the user using Wireshark tool.

The figure below highlights the location of the SNI field:

Figure 5: Server Name Indication Field

The figure below highlights the location of the CN field:

Figure 6: Common Name Field



How to Configure SSL Custom Application

Configuring SSL Custom Application

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar custom** *custom-protocol-name* **ssl unique-name** *regex* **id** *selector-id*
4. **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** *custom-protocol-name* **ssl unique-name** *regex* **id** *selector-id* **Example:** `Device (config)# ip nbar custom name ssl unique-name www.example.com id 11`	Defines the SSL-based custom protocol match and provides a hostname in the form of a regular expression. **Note** The hostname that is configured in this command is found either in the Server Name Indication (SNI) field in the Client Hello extensions or in the Common Name (CN) field in the digital certificate that the server sends to the client.
Step 4	**end** **Example:** `Router(config)# end`	(Optional) Exits global configuration mode.

Configuration Examples for the SSL Custom Application

Example: SSL Custom Applications

The following example displays how to configure SSL Custom Application. The hostname that is configured in this command is found either in the Server Name Indication (SNI) field in the Client Hello extensions or in the Common Name (CN) field in the digital certificate that the server sends to the client.

```
Device> enable
Device# configuration terminal
Device(config)# ip nbar custom name ssl unique-name  www.example.com id  11
Device(config)# exit
```

Additional References for SSL Custom Application

Related Documents for SSL Custom Application

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
SSL Sub-classification	NBAR Protocol Pack 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

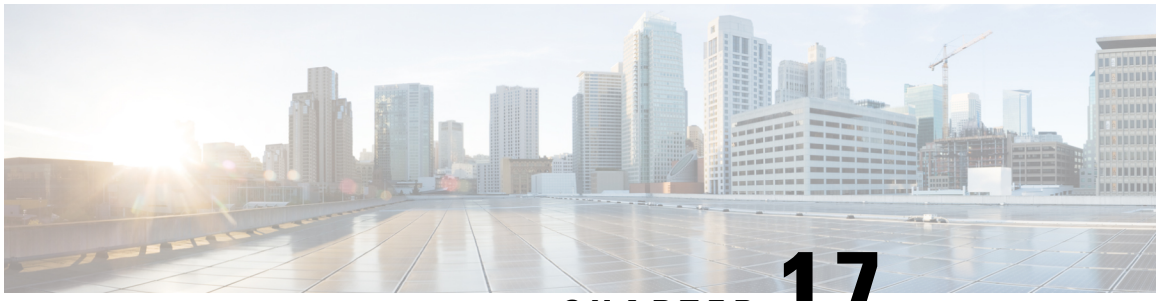
Feature Information for SSL Custom Application

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 20: Feature Information for SSL Custom Application

Feature Name	Releases	Feature Information
SSL Custom Application	Cisco IOS XE Release 3.15S	SSL Custom Application feature enables users to customize applications that run on any protocol over Secure Socket Layer (SSL), including HTTP over Secure Socket Layer (HTTPS), using the server name, if it exists in the Client Hello extensions, or the common name from the certificate that the server sends to the client. The following command was introduced or modified: **ip nbar custom**.



CHAPTER 17

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.

Feature Information

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 21: Feature Information for NBAR Fine-Grain Application Recognition Mode

Feature Name	Releases	Feature Information
Fine-grain application recognition mode	Cisco IOS XE Release 3.15S	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 Applications 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 coarse-grain mode and thus preventing any impact on the performance of the system. The following command was introduced or modified: **ip nbar custom**.
Fine-grain application recognition mode	Cisco IOS XE Release 3.16S Cisco IOS XE 16.x releases	Default mode changed to coarse-grain.

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".

Note For application attribute types, see Application Attributes, on page 197. For attribute values, see the protocol examples provided through the Protocol Library site.

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
```

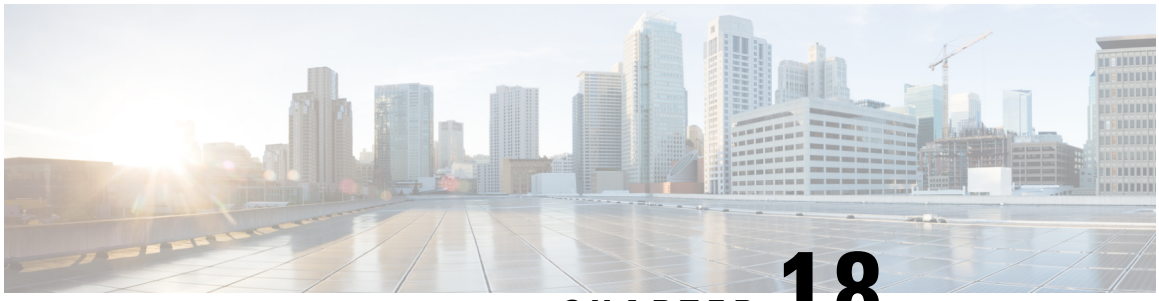
Additional References

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



CHAPTER 18

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.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

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

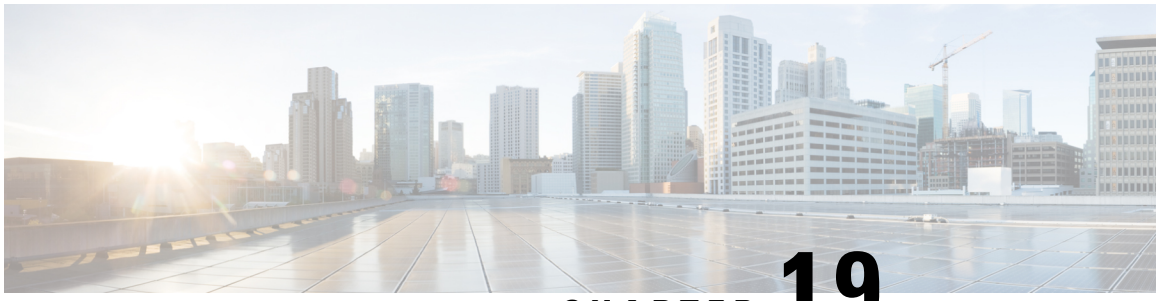
Feature Information for NBAR Custom Applications Based on DNS Name

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 22: Feature Information for NBAR Custom Applications Based on DNS Name

Feature Name	Releases	Feature Information
NBAR Custom Applications Based on DNS Name	Cisco IOS XE Release 3.15S	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 19

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 www.cisco.com/go/cfn. 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 164.
- 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

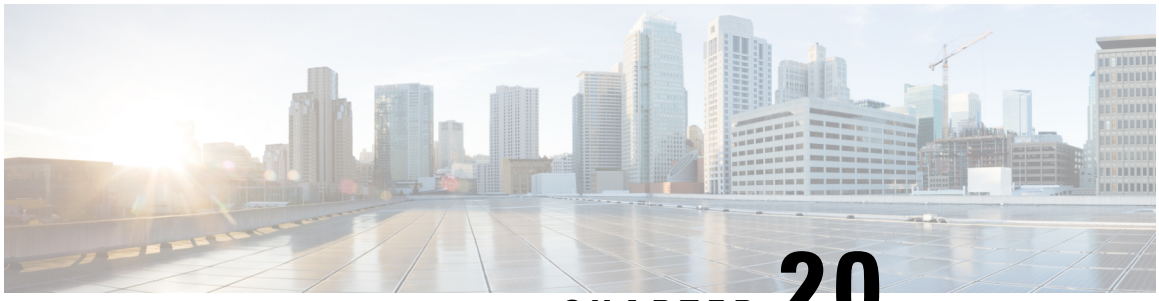
Feature Information for NBAR2 Auto-learn

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 23: 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 20

DNS-AS

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).

DNS-AS, "DNS as Authoritative Source," provides centralized control of custom application classification information.

This module contains concepts and tasks for configuring and using DNS-AS.

Introduction

Working together with Cisco NBAR2, "DNS as Authoritative Source," DNS-AS, provides centralized control of custom application classification information. Classification information (metadata such as application name, ID, traffic class, business relevance, and so on) is used by NBAR2 to recognize the network traffic of specific applications, and to classify the traffic by assigning parameters useful both in reporting and in applying network traffic policy.

Classification Metadata Reflects Organizational Needs, Policy Intent

Different enterprises have different requirements for reporting and shaping traffic through network traffic policy. This is partly because they use different local applications internal to the organization, and partly because widely used applications may have a different business relevance to different organizations.

Consequently, it is often helpful to customize application classification information to determine how network traffic is reported and shaped by traffic policy.

Leveraging DNS Infrastructure

DNS-AS leverages the universally available DNS query/response infrastructure to enable local DNS servers within an organization to propagate application classification information to routers in an enterprise network. The local DNS servers function as "authoritative source" for both DNS data and custom classification data.

Through its flexibility and simplicity, DNS-AS unlocks traffic reporting and shaping functionality that may otherwise be difficult to configure.

DNS-AS In Use

Setup

DNS-AS setup includes configuration steps on the local DNS server(s) and routers within the enterprise network.

Local DNS servers are configured with the classification information for specific "trusted domain" sites/applications. This enables a network administrator to control how a network handles traffic for these local, server-based applications - for example, those used in an enterprise intranet.

Routers are configured to detect DNS traffic for the "trusted domains" (sites/applications) controlled by DNS-AS.

Propagating Classification Information

When configuration is complete, the DNS servers can provide classification information for the "trusted domain" applications.

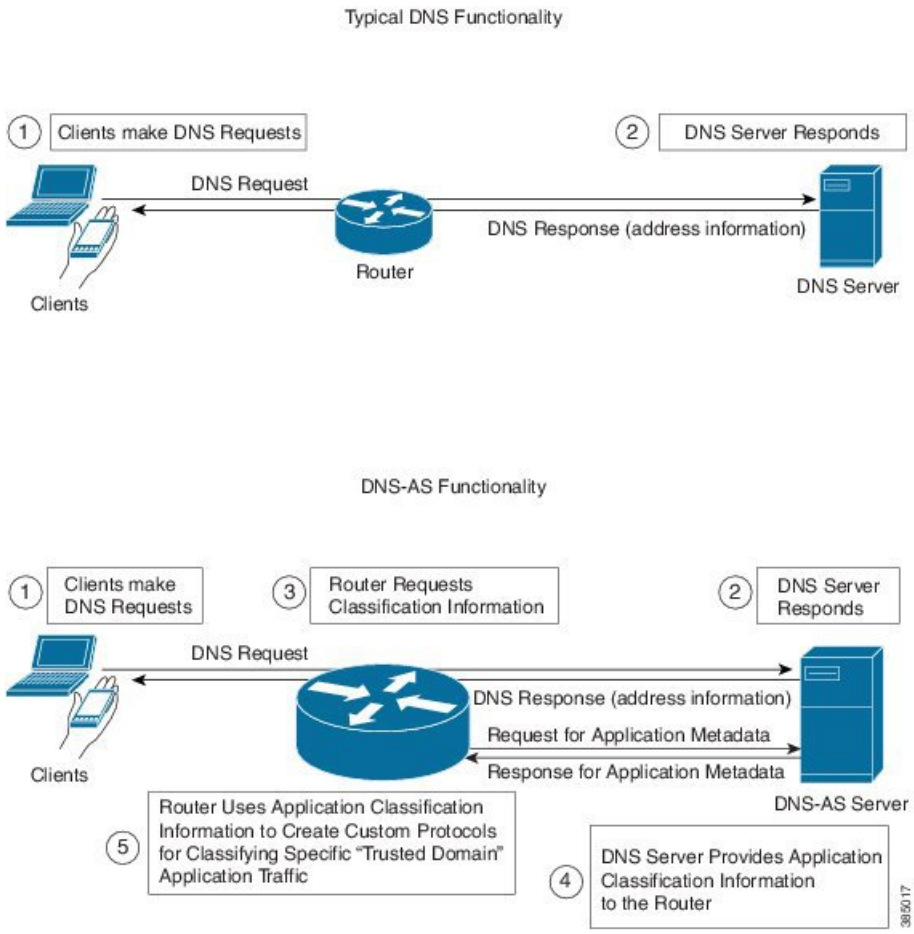
When a client in the network makes a DNS request, the DNS response is sent as usual. If the request relates to a "trusted domain" application, the router then queries the local DNS server about the application. The DNS server sends the router the network address data and the relevant classification information.

Using the Classification Information

On the routers that receive the information, NBAR2 uses the information to automatically create custom protocols that classify the traffic.

Traffic classification affects application visibility functions, such as reporting of traffic, and it affects application control, such as the use of QoS traffic-shaping policy.

Figure 7: DNS-AS Functionality



Priority

Custom application protocols have priority over standard Cisco NBAR2 Protocol Pack protocols, so it is possible to override Protocol Pack protocols by configuring application classification metadata using DNS-AS.

Predefined Protocols and Customized Protocols

For most network traffic, the predefined protocols in the NBAR2 Protocol Pack are sufficient to identify and classify traffic.

For local applications not covered by the Protocol Pack, such as the sites and applications in an enterprise's intranet, DNS-AS provides a centralized mechanism for controlling traffic.

Classification and Traffic Policy

Application classification and traffic policy are related but distinct. DNS-AS provides classification information, but does not directly control traffic policy.

Classification of applications may be controlled by:

- Cisco-provided Protocol Pack
- User-defined protocols
- Automatically-generated custom protocols
- DNS-AS-specified application metadata that indirectly creates custom protocols

Traffic policy may be defined by:

- Direct configuration of policy on the router
- Network controller, such as SDN, providing traffic policy

The following table clarifies the different types of protocols that can control classification of applications.

Table 24: Protocol Types that Control Application Classification

Protocol Type	Source
NBAR2 Protocol Pack protocols	Cisco-provided
Custom protocols defined manually	User-defined on router
Custom protocols defined automatically using DNS-AS	User-defined • Application metadata configured on DNS-AS server(s) • Trusted domains configured on router(s)

Efficient, Centralized Configuration

An advantage to using DNS-AS is efficiency of configuration. DNS-AS helps to control application classification over the entire enterprise network, but most of its configuration tasks are handled on the local DNS-AS server(s) operating within the network.

Configuration tasks include:

- **Configuring application metadata**: Defining the metadata for each “trusted domain” application during DNS-AS server setup; modifying the metadata at any time.
- **Configuring trusted domains**: Trusted domains are configured on the individual routers within the network.

Adaptability

Centralized configuration makes it easier to adapt to changes in the local applications. For example, if the IP addresses of the servers handling a local application change, or if the metadata attributes (application-class, business-relevance, and so on) for an application change, you can configure the changes on the local DNS-AS server(s) and the updates are propagated to the routers throughout the entire network.

DNS-AS vs. SDN Controller Functionality

DNS-AS and SDN controllers, when used, both operate broadly on the network. While an SDN controller provides traffic policy to devices in the network, DNS-AS provides application-match metadata.

NBAR2 Responding to Evolving Networks and Network Traffic

Applications Using End-to-End Encryption

Many of today's network applications operate in clear text over common transports such as HTTP. These applications can be identified using Deep Packet Inspection (DPI), a resource-intensive method. However, more and more network applications are communicating with end-to-end encryption, preventing identification by DPI.

Enterprise Networking Moving to the Cloud

A trend in enterprise networks is moving to the cloud. Instead of operating their own full-scale enterprise network, organizations are opting to move network infrastructure to cloud service providers. Their downsized internal network may have to control a variety of network devices located anywhere in the world. Those devices, managed by the cloud services provider, may not be under their direct administrative control.

NBAR Flexibility, Agility

Cisco NBAR2 features, such as DNS-AS, are evolving to address the changing trends in enterprise networking. While end-to-end encryption and migration to the cloud complicate the task of providing application visibility and control, NBAR continues to aim for:

- Simplicity in network configuration
- Agility at scale

Comparison with the Custom Protocol Feature

The DNS-AS configuration process is similar in some ways to using the NBAR2 **custom protocol** feature to create a protocol for a specific application relevant to the organization, but DNS-AS does not operate router-by-router for each individual application.

DNS-AS also provides an easier method of reconfiguring how numerous devices within the network handle custom applications.

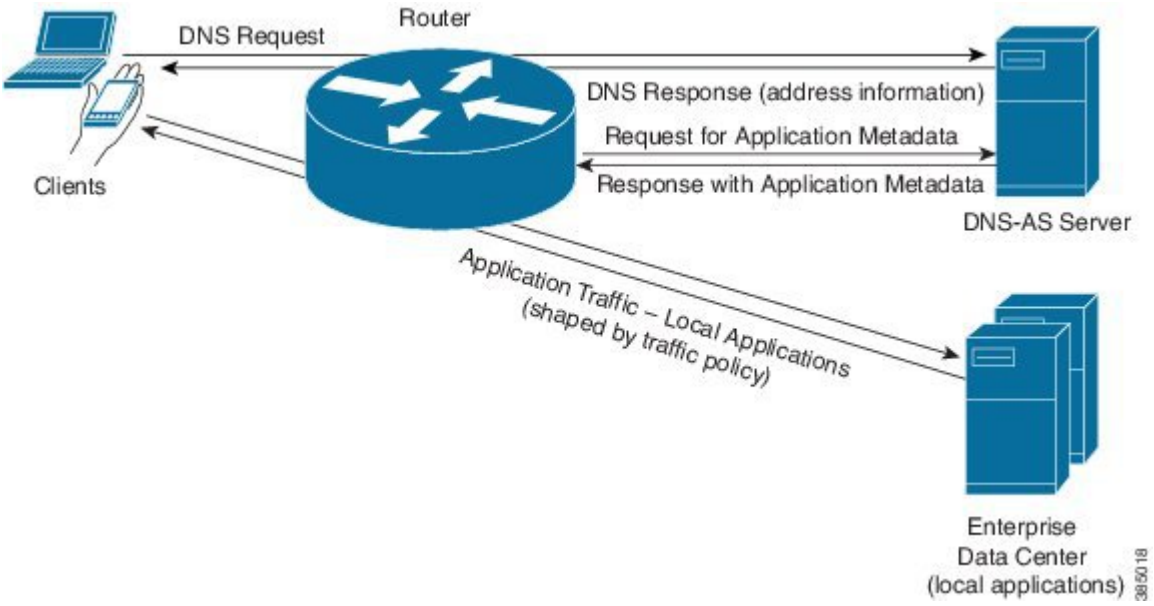
- When using the **custom protocol** feature, if any attributes of a custom application change, or if the server hosting the application changes, then updates to the custom application protocol must be made on each router in the network, one by one. In a network containing hundreds of routers, this process is impractical.
- When using **DNS-AS**, the single reconfiguration on the DNS server propagates information to all routers in the network.

DNS-AS Mechanism

Basic Topology

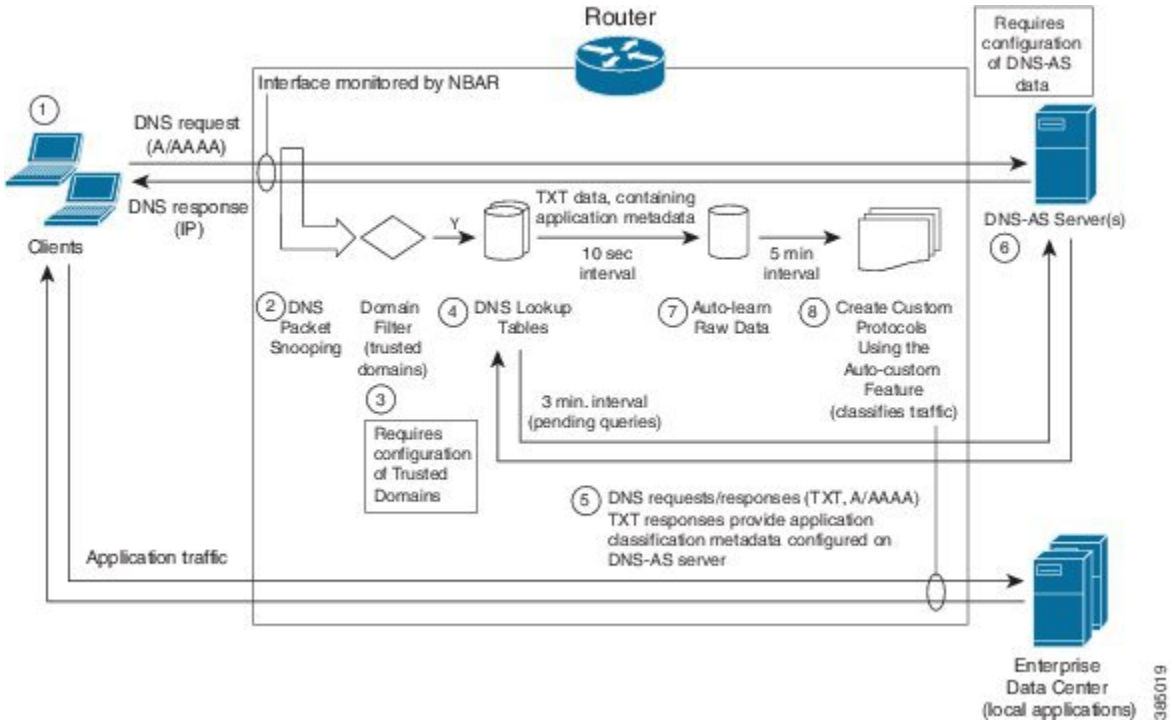
The following figure illustrates how the DNS-AS server operates with the clients (user devices), routers, and data centers (hosting local applications) within an enterprise network.

Figure 8: Topology



Mechanism Details

Figure 9: Details of Mechanism



1. A client (PC in the network) makes a request for a domain defined as a trusted domain. The request goes to the router to which the client is connected.

Example: A browser requests: StaffOnly.XYZ.com

2. Operating on the router, NBAR2 uses DNS packet snooping (on interfaces on which NBAR2 is active) to monitor the DNS requests from clients in the network. The router forwards the information in the DNS requests to the next step, which analyzes the data.

3. The Domain Filter mechanism identifies DNS requests for a trusted domain by matching to configured textual regular expressions.

Example regular expression: *staffonly.xyz.com

4. For each trusted domain identified in the previous step, the router prepares a "TXT" DNS request to send to the DNS-AS server. It collects these "pending" requests for a brief performance-optimizing interval.

5. The router server sends the "TXT" DNS requests to the DNS-AS server.

6. The DNS-AS server sends the "TXT" response. If the "TXT" response contains information with the "CISCO-CLS" prefix, the router sends an A request (requesting an IPv4 address) or an AAAA request (requesting an IPv6 address). The DNS-AS server sends an A or AAAA response to the request.

7. The router collects the "auto-learn raw data" responses containing application classification data for a brief interval.

8. Using the auto-learn raw data, NBAR creates custom application protocols for the relevant domains. Note: Custom application protocols have priority over standard Cisco NBAR2 Protocol Pack protocols.

DNS-AS Server Always Provides the Latest Version

If the router later makes a new request for a previously queried domain, the DNS-AS server sends the latest version of the metadata. So if the metadata has changed, the router will receive the new version.

DNS-AS Setup

DNS-AS requires configuration on local DNS servers and routers, as follows.

- DNS-AS Server Setup, on page 178
- DNS-AS Router Setup, on page 179

DNS-AS Server Setup

On local DNS servers within the enterprise network, configure application classification information for each "trusted domain." This is the information that the server propagates to routers when queried for application metadata. When the router sends a TXT query regarding an application, the DNS server sends the relevant metadata in the TXT response.

Application Metadata Fields

Application metadata is configured on the DNS-AS server(s). The individual routers in the network apply the metadata to create custom protocols and handle application traffic accordingly.

The following table describes the metadata fields that can be specified for applications handled by the DNS-AS feature. Customized application metadata specified using the DNS-AS feature has priority over any metadata provided by the NBAR2 Protocol Pack installed on a router.

Table 25: Metadata Fields

Field	Mandatory/Optional	Usage Notes
Application name (app-name)	Mandatory	For an application **included** in the installed Protocol Pack: Any customized metadata specified for the application takes priority over metadata specified in the Protocol Pack. For an application **not included** in the installed Protocol Pack: A new custom-protocol is created.
Application ID (app-id)	Mandatory	If not specified, NBAR2 generates an application-id. Not valid for existing applications. **Note** It is recommended to include this field when specifying a new application. This provides a universal ID number for the customized application within the network. The universal ID number enables traffic data collectors to aggregate DNS-AS custom application classification data coming from different devices within the network.

Field	Mandatory/Optional	Usage Notes
Traffic class (app-class)	Optional	If this field is specified and the business-relevance field is not specified, NBAR2 automatically assigns the business-relevance field a value of "business-relevant". **Note** It is strongly recommended to include this field when specifying a new application. Without specifying traffic-class, the application uses the default traffic-class value.
Business relevance (business)	Optional	Business relevance

DNS-AS Router Setup

On Cisco routers operating in the network, activate the DNS-AS feature and configure the DNS-AS server(s) to use, as well as the "trusted domains," as follows:

Step 1: Activate DNS-AS on Routers in the Network

On the routers in the network, activate DNS-AS.

```
avc dns-as client enable
```

Step 2: Specify the DNS-AS Server(s) to Use

Specify the DNS-AS server(s) to query with TXT requests for classification metadata.

```
ip name-server vrf <name> <address>
```

For details, see Configuring the DNS-AS Server for a Router to Query, on page 181.

Step 3: Configure Trusted Domains on Routers in the Network

On the routers in the network, configure "trusted domains." The DNS-AS feature affects only the applications configured as trusted domains.

When a router detects DNS traffic for a trusted domain, it requests and receives application classification metadata from the local DNS-AS server using TXT request/response.

Configure trusted domains by providing textual regular expressions that will match domain names found in DNS requests sent by clients in the network. For the example above, **StaffOnly.XYZ.com**, the regular expression might be:

```
*staffonly.xyz
```

For details, see Configuring Trusted Domains, on page 182.

Deploying a New Application in the Network

When deploying a new local application in the organization's network, review the procedures for setting up DNS-AS to add the new application to the DNS-AS server setup and the router "trusted domain" setup.

Restrictions

The following restrictions apply to using DNS-AS:

- Only IPv4 DNS servers are supported.
- Maximum of 50 DNS-AS custom-applications are supported (across all DNS servers within the network).

Note NBAR2 supports a total of 120 custom protocols. Custom protocols generated by DNS-AS count toward the total.

- Maximum of 2 VRFs are supported.
- Maximum of 2 servers are supported per VRF.
- NBAR performs DNS packet snooping only on DNS traffic on interfaces on which NBAR is configured.
- A DNS-AS custom-application protocol can include either IPv4 addresses or IPv6 addresses, but not both.
- When using DNS-AS to customize existing applications, the "app-id" field should either be omitted from TXT record or be identical to the exiting application "app-id".
- For applications that are not included in the NBAR2 Protocol Pack, the "app-name" field must be unique across all TXT records across all DNS-AS servers.

DNS-AS CLI Commands

Several CLIs are used on routers in the network to configure and monitor DNS-AS.

See the following sections:

- Activating and Configuring DNS-AS, on page 180
- Monitoring DNS-AS, on page 185

Activating and Configuring DNS-AS

The following reference table provides a summary of DNS-AS configuration commands.

Table 26: Configuration Commands

CLI	Description
ip name-server vrf *name address*	Configures the DNS server. When used for other router functions, this CLI can support several VRFs, and up to 6 IP addresses per VRF. However, DNS-AS supports only 2 VRFs, and the first 2 servers configured per VRF. **Usage Notes**: • The specified VRF does not have to be defined at the time that the CLI is executed. • Configuration of more than one server is used for redundancy or VRF. • Immediately after configuration, DNS-AS prioritizes the configured servers in the order in which they were configured. After restarting the router, DNS-AS prioritizes in alphabetical order, using the VRF name. • You can view the configuration using the **show avc dns-as client name-server brief** command. This indicates which servers the DNS-AS feature is using.

1. **avc dns-as client trusted-domains**
2. **domain** *regular-expression*
	Configures trusted domains, using a regular expression as a filter.
	Usage Notes:
	Can configure up to 50 trusted domains.
	Example:
	`Device(config)#avc dns-as client trusted-domains`
	`Device(config-trusted-domains)#domain *staffonly.xyz.com`
	`Device(config-trusted-domains)#exit`

avc dns-as client enable	Enables DNS-AS.

1. **interface** *interface*
2. **avc dns-as learning**
	Enables NBAR on an interface.
	The DNS-AS feature is only active on interfaces monitored by NBAR. If NBAR has been activated on an interface for use with any NBAR feature, the interface will be monitored for DNS-AS also.
	Example:
	`Device#config terminal`
	`Device(config)#interface gig 0/0/0`
	`Device(config-if)#avc dns-as learning`
	`Device(config-if)#exit`

Configuring the DNS-AS Server for a Router to Query

Use the following procedure on a router to configure the DNS-AS server(s). For information about displaying the configured DNS servers, see Displaying Active DNS Servers, on page 189.

SUMMARY STEPS

1. **configure terminal**
2. **ip name-server vrf** *name address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 2	**ip name-server vrf** *name address* **Example:** This example specifies a DNS server called MANAGEMENT. `Device(config)#ip name-server vrf MANAGEMENT 10.56.56.56`	Specifies the DNS server.

Configuring Trusted Domains

The DNS-AS feature operates only on applications configured in the trusted domain list.

Configure trusted domains by specifying regular expressions to match the domain name—for example, ***cisco.com** for all Cisco.com traffic, including **www.cisco.com** and **developer.cisco.com**.

When specifying trusted domains, it may be helpful to use a packet analyzer application, such as the open-source Wireshark application, to examine DNS request packets for trusted applications. The domain name appears in the packet and can be used for building effective regular expressions.

Use the following procedure on a router to configure a new trusted domain.

SUMMARY STEPS

1. **configure terminal**
2. **avc dns-as client trusted-domains**
3. **domain** *regular-expression*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 2	**avc dns-as client trusted-domains** **Example:**	Enters trusted domain configuration mode.

	Command or Action	Purpose
	`Device(config)#avc dns-as client trusted-domains`	
Step 3	**domain** *regular-expression* **Example:** `Device(config-trusted-domains)#domain *staffonly.xyz.com`	The regular expression specifies the domain.

Enabling DNS-AS

Use the following procedure on a router to enable the DNS-AS feature.

SUMMARY STEPS

1. **configure terminal**
2. **avc dns-as client enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 2	**avc dns-as client enable** **Example:** `Device(config)#avc dns-as client enable`	Enables the DNS-AS feature.

Disabling DNS-AS

Use the following procedure on a router to disable the DNS-AS feature.

SUMMARY STEPS

1. **configure terminal**
2. **no avc dns-as client enable**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**no avc dns-as client enable** **Example:** `Device(config)#no avc dns-as client enable`	Disables the DNS-AS feature.

Enabling NBAR on an Interface for DNS-AS

When using DNS-AS, each router in the network must snoop DNS traffic from clients in the network and forward the data to the next step of the DNS-AS process, the domain filter.

For the router to monitor the DNS requests, NBAR must be enabled on the interfaces on which the router receives DNS requests from clients. As a general rule, the router monitors DNS traffic on all interfaces on which NBAR is enabled.

Numerous CLIs can enable NBAR on an interface. When using DNS-AS, use the following procedure to enable NBAR on the interface for DNS-AS learning.

Note In cases where NBAR is already enabled on the interface, this task is redundant. For example, if IP protocol discovery is already enabled on the interface, the procedure is not necessary. However, for clarity, even in these redundant situations, it is recommended to use this procedure.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface*
3. **avc dns-as learning**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface* **Example:** `Device(config)#interface gig 0/0/0`	Enter interface configuration mode for a specific interface.
Step 3	**avc dns-as learning** **Example:** `Device(config-if)#avc dns-as learning`	Enable NBAR on the interface specified in a previous step.

Monitoring DNS-AS

The following reference table provides a summary of DNS-AS monitoring commands.

Table 27: Monitoring Commands

CLI	Description
show avc dns-as client statistics	Show receive/transmit counters per server.
show avc dns-as client binding-table	Show DNS-AS custom-application data.
show avc dns-as client binding-table detailed	Show DNS-AS custom-application data in a record format.
clear avc dns-as client statistics	Clear the receive/transmit counters.
show avc dns-as client name-server brief	Show configured DNS servers.
show ip nbar classification auto-learn dns-as *<1-100>*	Show the auto-learn table. The number specifies the number of entries to display in the table.
clear ip nbar classification auto-learn dns-as-client statistics	Clear the auto-learn table.
clear ip nbar classification auto-learn dns-as restart	Restart all DNS-AS learning. All databases are cleared.
show ip nbar classification auto-learn dns-as pending-queries	Show pending queries.
clear ip nbar classification auto-learn dns-as pending-queries	Clear pending-queries statistics. **Usage Notes**: This CLI does not cause injection of pending queries.
show ip nbar protocol-discovery stats packet-count	Display the packet count for all NBAR protocols, including the custom protocols generated by DNS-AS.

Showing DNS-AS Client Statistics

Use this procedure to display DNS-AS client statistics. The results display the running total of number of packets, and are displayed per server.

Usage:

- Disabling DNS-AS resets the statistics.
- In some cases, unusually high traffic volume may cause some statistics to fail, in which case the command output displays "Error" for some statistics.

SUMMARY STEPS

1. **show avc dns-as client statistics**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show avc dns-as client statistics**
	Example:
	`Device#show avc dns-as client statistics`
	`csi-mcp-asr1k-02#show avc dns-as client statistics`
	`Server details: vrf-id = 2 vrf-name = MNG ip = 10.56.196.50`
	`AAAA Query Error packets 0`
	`AAAA Query TX packets 0`
	`AAAA Response RX packets 0`
	`TXT Query Error packets 0`
	`TXT Query TX packets 50`
	`TXT Response RX packets 50`
	`A Query Error packets 0`
	`A Query TX packets 50`
	`A Response RX packets 50`
	`Total Drop packets 0`
	`Server details: vrf-id = 5 vrf-name = vrf2 ip = 10.56.196.51`
	`AAAA Query Error packets 0`
	`AAAA Query TX packets 0`
	`AAAA Response RX packets 0`
	`TXT Query Error packets 0`
	`TXT Query TX packets 0`
	`TXT Response RX packets 0`
	`A Query Error packets 0`
	`A Query TX packets 0`
	`A Response RX packets 0`
	`Total Drop packets 0`
		Display client statistics, per server.

Showing the DNS-AS custom-application Data

Use this procedure to display DNS-AS custom-application data in **binding table format**. Also see the **detailed** form of the command, which presents the same information in **record format**, which enables piping the data into another application. See Showing the DNS-AS custom-application Data – Detailed, on page 187.

The information includes:

- Maximum number of protocols that can be customized using DNS-AS.
- Customization interval—Interval during which the router collects auto-learn raw data before creating new custom protocols. Default: 5 minutes
- Table of protocols currently stored in the binding table, with the VRF name, server IP, age, metadata, TTL, and Time to Expire data.

When Do Protocols Reach This Table?

The DNS-AS process has built-in rate limiters that introduce short delays to optimize overall performance. The major intervals that affect when protocols appear in the binding table are (total of about 8 minutes by default):

- Rate limiter before router sends DNS request to the DNS server (default: 3 minutes).
- Rate limiter after the router receives a DNS response from the DNS server (default: 10 seconds).
- Rate limiter before collected raw data is used to generate custom-protocols (default: 5 minutes).

SUMMARY STEPS

1. **show avc dns-as client binding-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show avc dns-as client binding-table** **Example:** `Device#show avc dns-as client binding-table`	Displays a binding table populated by custom-application data.

Showing the DNS-AS custom-application Data – Detailed

Use this procedure to display DNS-AS custom-application data in a record format, which enables piping the output into another application.

This procedure uses a command identical to the one described for the Showing the DNS-AS custom-application Data, on page 186 procedure, but with addition of the **detailed** keyword:

show avc dns-as client binding-table detailed

The following example uses the sec command in a UNIX-like environment to select output for the xyz domain. The command output is piped to sec, which filters for **staffonly**.

```
Device#show avc dns-as client binding-table detailed | sec staffonly
Protocol-name        : staffonly
 VRF                 : MNG
 Host                : staffonly.xyz.com
 Age[min]            : 17
 TTL[min]            : 1440
 Time to Expire[min] : 1420
 TXT Record          : app-name:staffonly|app-class:BULK-DATA
 IP                  : 10.2.3.10
```

Clearing the Receive and Transmit Counters

Use this procedure to clear the receive and transmit counters for the DNS-AS client statistics.

SUMMARY STEPS

1. **clear avc dns-as client statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear avc dns-as client statistics** **Example:** `Device#clear avc dns-as client statistics`	Clears receive and transmit counters.

Clearing the auto-learn Table

Use this procedure to clear the auto-learn raw data.

The auto-learn raw data is collected in the control plane for an interval (default 5 minutes) before being sent to the mechanism that creates custom protocols based on the data. Shortly after being cleared (typically within 10 seconds), the table is regenerated when the same data or a subset of the data, is sent again from the data plane, where the data typically has a 24-hour TTL, back to the auto-learn raw data table in the control plane.

By contrast, using the Clearing and Restarting DNS-AS Learning, on page 188 procedure clears the auto-learn raw data and any custom-protocols that have been created.

SUMMARY STEPS

1. **clear ip nbar classification auto-learn dns-as-client statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip nbar classification auto-learn dns-as-client statistics** **Example:** `Device#clear ip nbar classification auto-learn dns-as-client statistics`	Clears the auto-learn raw data.

Clearing and Restarting DNS-AS Learning

Use this command to clear the auto-learn raw data and any custom protocols that have been generated. All databases are cleared and the auto-learn process restarts without any prior data.

SUMMARY STEPS

1. **clear ip nbar classification auto-learn dns-as restart**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip nbar classification auto-learn dns-as restart** **Example:** `Device#clear ip nbar classification auto-learn dns-as restart`	Clears the auto-learn raw data and any custom protocols that have been generated.

Displaying Active DNS Servers

Use this procedure to display the DNS servers configured to operate with DNS-AS. For information about configuring DNS servers, see Configuring the DNS-AS Server for a Router to Query, on page 181.

SUMMARY STEPS

1. **show avc dns-as client name-server brief**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show avc dns-as client name-server brief** **Example:** `Device#show avc dns-as client name-server brief`	Displays the DNS servers configured to operate with DNS-AS.

Showing DNS-AS Auto-learn Data

Use this procedure to display the data collected in the DNS-AS auto-learn raw data repository before it has been used to create custom protocols.

For information about clearing the auto-learn data, see Clearing the auto-learn Table, on page 188.

When Does Data Reach This Table?

The DNS-AS process has built-in rate limiters that introduce short delays to optimize overall performance. The major intervals that affect when data reaches the auto-learn step are (total of about 3 minutes by default):

- Rate limiter before router sends DNS request to the DNS server (default: 3 minutes).
- Rate limiter after the router receives a DNS response from the DNS server (default: 10 seconds).

SUMMARY STEPS

1. **show ip nbar classification auto-learn dns-as-client** <*1-100*> **detailed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ip nbar classification auto-learn dns-as-client** <*1-100*> **detailed** **Example:** `Device#show ip nbar classification auto-learn dns-as-client 100 detailed`	Displays the data collected in the DNS-AS auto-learn raw data repository. The number determines how many entries to display in the table.

Displaying Pending DNS Queries

Use this procedure to display the DNS queries that the router has not yet sent to the DNS server. A rate limiter limits transmission of accumulated DNS queries to the DNS-AS server to an interval of 3 minutes. This optimizes system performance by not overloading the DNS-AS server with many identical requests.

See the Clearing the Pending DNS Query Statistics, on page 190 procedure for clearing the pending queries statistics.

SUMMARY STEPS

1. **show ip nbar classification auto-learn dns-as pending-queries**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show ip nbar classification auto-learn dns-as pending-queries**
	Example:
	`Device#show ip nbar classification auto-learn dns-as pending-queries`
	`AAAA queries pending inject 0`
	`AAAA queries injected 0`
	`TXT queries pending inject 0`
	`TXT queries injected 50`
	`A queries pending inject 0`
	`A queries injected 50`
		Display the pending queries.

Clearing the Pending DNS Query Statistics

Use this procedure to clear the pending queries statistics. See the Displaying Pending DNS Queries, on page 190 procedure for displaying the statistics.

SUMMARY STEPS

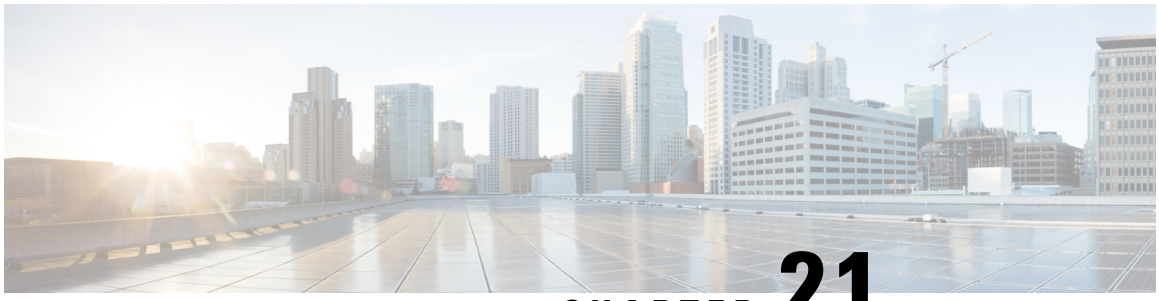
1. **clear ip nbar classification auto-learn dns-as pending-queries**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ip nbar classification auto-learn dns-as pending-queries** **Example:** `Device#clear ip nbar classification auto-learn dns-as pending-queries`	Clear the pending query statistics.

DNS-AS Troubleshooting

For DNS-AS Troubleshooting, see Cisco DNS-AS Troubleshooting.



CHAPTER 21

DNS Protocol Classification Change

Traffic for a network application includes DNS query/response traffic and the actual application flow. Using the DNS Protocol Classification Change feature, NBAR2 can be configured to classify and handle DNS traffic in the same way as its associated application traffic.

This module describes DNS Protocol Classification Change and the how to enable it.

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for DNS Protocol Class Change

None.

Information About DNS Protocol Classification Change

DNS Protocol Classification Change

Traffic for a network application includes DNS query/response traffic and the actual application flow. When classifying traffic, most attention is given to the application flow, both for reporting (application visibility) and control (QoS policy).

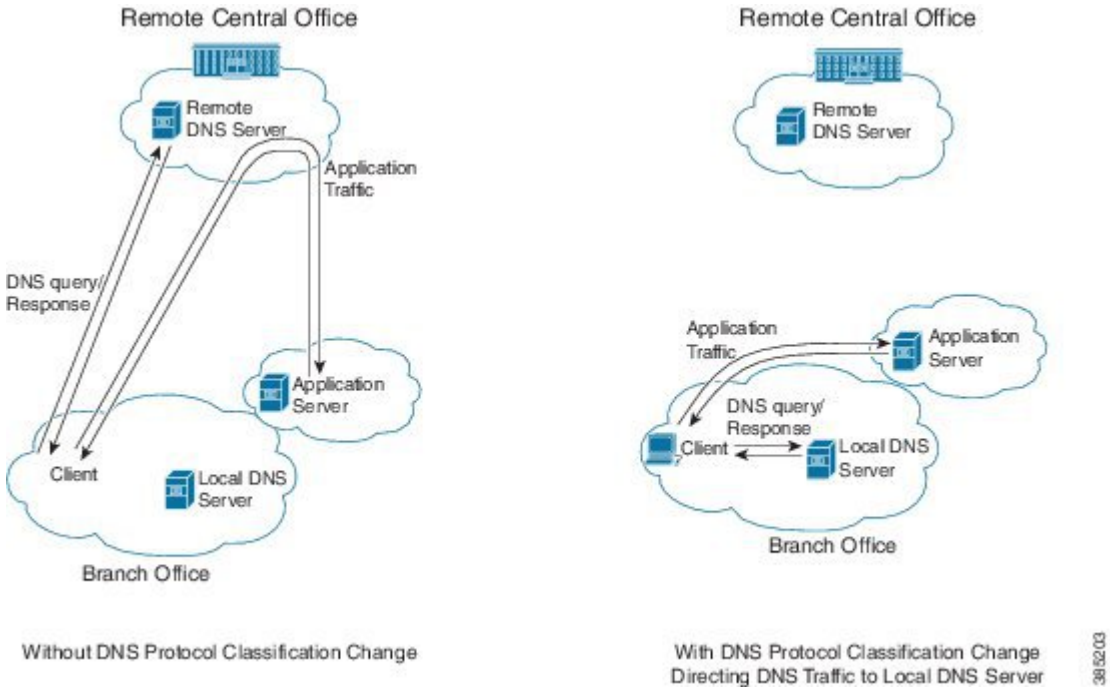
The DNS Protocol Classification Change feature enables an NBAR client, such as a router, to classify and handle DNS traffic in the same way as its associated application traffic. This is accomplished using the domain name that appears in the DNS flow.

Use of DNS Protocol Classification Change

DNS Protocol Classification Change can be especially useful in networks employing Cisco Intelligent WAN (IWAN), for optimizing the performance of network applications.

For example, in an IWAN spanning a wide geography, it might happen that a specific type of application traffic (example: Microsoft Office 365) may be routed first to a geographically distant node in the IWAN, and then to the relevant server. This route may diminish performance of the application. Using DNS protocol classification change, it is possible to redirect the DNS query/response to a local DNS server, and route the application traffic directly to the relevant cloud-based application server, improving application performance.

Figure 10: DNS Protocol Classification Change Improving Application Performance in an IWAN Environment



Usage Notes

- DNS Protocol Classification Change classifies the DNS flow in the same way as the application, based on built-in protocols or custom signatures.
- The DNS flow classification inherits the attributes of the application – category, business-relevance, traffic-class, encryption, and so on. For example, for a DNS flow classified as “Google-accounts” the encryption attribute is TRUE.
- DNS flows are not cached using the socket cache mechanism.
- To catch all DNS traffic for QoS, use the following “transport hierarchy” CLI:

  match protocol dns in-app-hierarchy

- Default: enabled.

How to Enable DNS Protocol Classification Change

Enabling DNS Protocol Classification Change

Enabling the DNS Protocol Classification Change feature enables an NBAR client, such as a router, to classify and handle DNS traffic in the same way as its associated application traffic.

The **no** form of the command disables the feature.

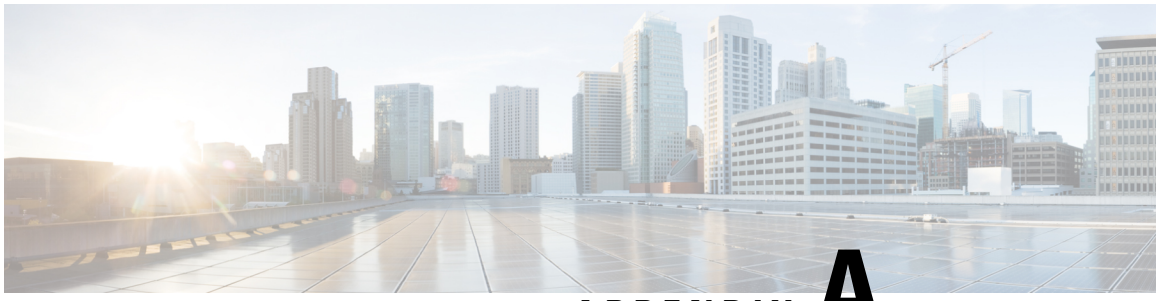
[**no**] **ip nbar classification dns classify-by-domain**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nbar classification dns classify-by-domain**

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 classification dns classify-by-domain** **Example:** `Device(config)# ip nbar classification dns classify-by-domain`	Enables the DNS Protocol Classification Change feature.



APPENDIX A

Application Attributes

About Attributes

The information that NBAR2 uses to recognize and classify application traffic is organized as application protocols. Each protocol has a set of attributes that relate to the specific network application. The list of attribute types are provided here.

Attribute Types

	Attribute	Description
Categorization	application-family	Categorization of the application: scheme 1.
	application-set	Categorization of the application: scheme 2.
	category	Categorization of the application: scheme 3.
	sub-category	Application usage.
Service	application-group	Group of applications that belong to the same service.
Priority	business-relevance	Indicates business-oriented applications.
Traffic attributes	encrypted	Possible encrypted traffic.
	p2p-technology	Peer-to-peer traffic.
	traffic-class	Application class-of-service (based on RFC 4594).
	tunnel	Tunnel-related traffic.