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

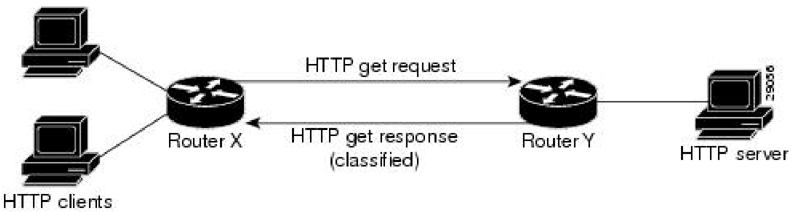
Classification of HTTP Traffic by a URL Host or MIME

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 18).

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 19).

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 10.

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 18.

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 22)

  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 23)

  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 22.

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 30)

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
```

```
             tunel :  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 16
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.