Cisco Umbrella

You can configure the device to redirect DNS requests to Cisco Umbrella, so that your FQDN policy defined in Cisco Umbrella can be applied to user connections. The following topics explain how to configure the Umbrella Connector to integrate the device with Cisco Umbrella.

About Cisco Umbrella Connector

If you use Cisco Umbrella, you can configure the Cisco Umbrella Connector to redirect DNS queries to Cisco Umbrella. This allows Cisco Umbrella to identify requests to unpermitted or questionable domain names and apply your DNS-based security policy.

The Umbrella Connector is part of the system's DNS inspection. If your existing DNS inspection policy map decides to block or drop a request based on your DNS inspection settings, the request is not forwarded to Cisco Umbrella. Thus, you have two lines of protection: your local DNS inspection policy and your Cisco Umbrella cloud-based policy.

When redirecting DNS lookup requests to Cisco Umbrella, the Umbrella Connector adds an EDNS (Extension mechanisms for DNS) record. An EDNS record includes the device identifier information, organization ID, and client IP address. Your cloud-based policy can use those criteria to control access in addition to the reputation of the FQDN. You can also elect to encrypt the DNS request using DNSCrypt to ensure the privacy of usernames and internal IP addresses.

Cisco Umbrella Enterprise Security Policy

In your cloud-based Cisco Umbrella Enterprise Security policy, you can control access based on the reputation of the fully-qualified domain name (FQDN) in the DNS lookup request. Your Enterprise Security policy can enforce one of the following actions:

- Allow—If you have no block rules for an FQDN, and Cisco Umbrella determines that it belongs to a non-malicious site, then the site's actual IP address is returned. This is normal DNS lookup behavior.

- Proxy—If you have no block rules for an FQDN, and Cisco Umbrella determines that it belongs to a suspicious site, then the DNS reply returns the IP address of the Umbrella intelligent proxy. The proxy can then inspect the HTTP connection and apply URL filtering. You must ensure that intelligent proxy is enabled from the Cisco Umbrella dashboard (**Security Setting** > **Enable Intelligent Proxy**).
- Block—If you explicitly block an FQDN, or Cisco Umbrella determines that it belongs to a malicious site, then the DNS reply returns the IP address of the Umbrella cloud landing page for blocked connections.

Cisco Umbrella Registration

When you configure the Umbrella Connector on a device, it registers with Cisco Umbrella in the cloud. The registration process assigns a single device ID, which identifies one of the following:

- One standalone device in single context mode.
- One high availability pair in single context mode.
- One cluster in single context mode.
- One security context in a multiple-context standalone device.
- One security context of a high availability pair
- One security context of a cluster.

Once registered, the device details will appear on the Cisco Umbrella dashboard. You can then change which policy is attached to a device. During registration, either the policy you specify in the configuration is used, or the default policy is assigned. You can assign the same Umbrella policy to multiple devices. If you specify the policy, the device ID you receive differs from what you would get if you did not specify a policy.

Switching from Legacy API Token to API/Secret Keys

Umbrella changed the API registration mechanism from an API token to an API key and Secret key starting with ASA 9.23(1). If you are using legacy tokens, you can switch from the legacy API to the API/Secret key API by doing the following. See the related procedures for detailed steps.

Procedure

Step 1 Import the certificate for the new API. You need to install the ISRG Root X1 self-signed certificate.

Step 2 Disable Umbrella.

Step 3 Remove the existing token registration.

Step 4 Configure the new API and secret keys.

Step 5 Re-enable Umbrella.

Licensing Requirements for Cisco Umbrella Connector

To use the Cisco Umbrella Connector, you must have a 3DES license. If you are using Smart Licensing, your account must be enabled for export-controlled functionality.

The Cisco Umbrella portal has separate licensing requirements.

Guidelines and Limitations for Cisco Umbrella

Context Mode

- In multiple-context mode, you configure the Umbrella Connector in each context. Each context has a separate device ID, and is represented as a separate device in the Cisco Umbrella Connector dashboard. The device name is the hostname configured in the context, plus the hardware model, plus the context name. For example, CiscoASA-ASA5515-Context1.

Failover

- The active unit in the high availability pair registers the pair as a single unit with Cisco Umbrella. Both peers use the same device ID, which is formed from their serial numbers: *primary-serial-number_secondary-serial-number*. For multiple context mode, each pair of security contexts is considered a single unit. You must configure high availability, and the units must have successfully formed a high-availability group (even if the standby device is currently in a failed state), before enabling Cisco Umbrella, or the registration will fail.

Cluster

- The cluster control unit registers the cluster as a single unit with Cisco Umbrella. All peers use the same device ID. For multiple context mode, a security context in the cluster is considered a single unit across all peers.

Additional Guidelines

- Redirection to Cisco Umbrella is done for DNS requests in through traffic only. DNS requests that the system itself initiates are never redirected to Cisco Umbrella. For example, FQDN-based access control rules are never resolved based on Umbrella policy, nor are any FQDNs that are used in other commands or configuration settings.
- The Cisco Umbrella Connector works on any DNS request in through traffic. However, the block and proxy actions are effective only if the DNS response is then used for HTTP/HTTPS connections, because the IP address returned is for a web site. Any blocked or proxied addresses for non-HTTP/HTTPS connections will either fail or complete in a misleading fashion. For example, pinging a blocked FQDN would result in pinging the server that hosts the Cisco Umbrella cloud block page.

Note Cisco Umbrella does try to intelligently identify FQDNs that might be non-HTTP/HTTPS, and does not return the IP address to the intelligent proxy for those FQDNs for proxied domain names.

- The system sends DNS/UDP traffic only to Cisco Umbrella. If you enable DNS/TCP inspection, the system does not send any DNS/TCP requests to Cisco Umbrella. However, DNS/TCP requests do not increment the Umbrella bypass counter.
- If you enable DNScrypt for Umbrella inspection, the system uses UDP/443 for the encrypted session. You must include UDP/443 along with UDP/53 in the class map that applies DNS inspection for Cisco Umbrella for DNScrypt to work correctly. Both UDP/443 and UDP/53 are included in the default inspection class for DNS, but if you create a custom class, ensure that you define an ACL that includes both ports for the match class.
- DNScrypt uses IPv4 only for the certificate update handshake. However, DNSscrypt does encrypt both IPv4 and IPv6 traffic.
- There must be an IPv4 route to the Internet that can reach api.umbrella.com and api.opendns.com (registration uses IPv4 only). You also must have routes to the following DNS resolvers, and your access rules must allow DNS traffic to these hosts. These routes can go through either the data interfaces or the management interface; any valid route will work for both registration and DNS resolution. The default servers that the system uses are indicated; you can use the other servers by configuring the resolver in the Umbrella global settings.
  - 208.67.220.220 (system default for IPv4)
  - 208.67.222.222
  - 2620:119:53::53 (system default for IPv6)
  - 2620:119:35::35
- The system does not support the Umbrella FamilyShield service. If you configure the FamilyShield resolvers, you might get unexpected results.
- When evaluating whether to fail open, the system considers whether the Umbrella resolver is down, or if an intervening device drops the DNS request or response based on how long it has waited for the response after sending out the request. Other factors, such as no route to the Umbrella resolver, are not considered.
- To unregister a device, first delete the Umbrella configuration, then delete the device from the Cisco Umbrella dashboard.
- Any web requests that use IP addresses instead of FQDN will bypass Cisco Umbrella. In addition, if a roaming client obtains DNS resolution from a different WAN connection than the one that goes through an Umbrella-enabled device, connections that use those resolutions bypass Cisco Umbrella.
- If a user has an HTTP proxy, then the proxy might be doing DNS resolution, and the resolutions will not go through Cisco Umbrella.
- NAT DNS46 and DNS64 are not supported. You cannot translate DNS requests between IPv4 and IPv6 addressing.
- The EDNS record will include both the IPv4 and IPv6 host addresses.

- If the client uses DNS over HTTPS, then the cloud security service will not inspect DNS and HTTP/HTTPS traffic.

Configure Cisco Umbrella Connector

You can configure the device to interact with Cisco Umbrella in the cloud. The system redirects DNS lookup requests to Cisco Umbrella, which then applies your cloud-based Enterprise Security fully-qualified domain name (FQDN) policy. For malicious or suspicious traffic, users can be blocked from a site, or redirected to an intelligent proxy that can perform URL filtering based on your cloud-based policy.

The following procedure explains the end-to-end process for configuring the Cisco Umbrella Connector.

Before you begin

In multiple-context mode, perform this procedure in each security context that should use Cisco Umbrella.

Procedure

Step 1 Establish an account on Cisco Umbrella, https://umbrella.cisco.com.

Step 2 Install the CA Certificate from the Cisco Umbrella Registration Server, on page 6.

The device registration uses HTTPS, which requires that you install the root certificate.

Step 3 If it is not already enabled, configure DNS servers and enable DNS lookup on the interfaces.

You can use your own servers, or configure the Cisco Umbrella servers. DNS inspection automatically redirects to the Cisco Umbrella resolvers even if you configure different servers.

- 208.67.220.220
- 208.67.222.222
- 2620:119:53::53
- 2620:119:35::35

Example:

```
ciscoasa(config)# dns domain-lookup outside
ciscoasa(config)# dns domain-lookup inside
ciscoasa(config)# dns name-server 208.67.220.220
```

Step 4 Configure the Umbrella Connector Global Settings, on page 7.

Step 5 Enable Umbrella in the DNS Inspection Policy Map, on page 9.

Step 6 Verify the Umbrella Registration, on page 10.

Install the CA Certificate from the Cisco Umbrella Registration Server

You must import the root certificate to establish the HTTPS connection with the Cisco Umbrella registration server. The system uses the HTTPS connection when registering the device. In Cisco Umbrella choose **Deployments > Configuration > Root Certificate** and download the certificate.

When using the API and secret keys to register to Umbrella, you need to install the the ISRG Root X1 self-signed certificate pem file. You can obtain this certificate from https://letsencrypt.org/certs/isrgrootx1.pem.

Before you begin

When Umbrella updates its certificate, you need to download the new certificate. The root certificate might also change. Ensure that you have the correct root certificate uploaded.

When you update the certificate, you must disable Umbrella, then enable it again, so the system picks up the new certificate and registers correctly with Umbrella.

Procedure

Step 1 Create a trustpoint for the Cisco Umbrella registration server.

crypto ca trustpoint *name*

You can use any name you want for the trustpoint (up to 128 characters), such as ctx1 or umbrella_server.

Example:

```
ciscoasa(config)# crypto ca trustpoint ctx1
ciscoasa(config-ca-trustpoint)#
```

Step 2 Indicate that you want to manually enroll by pasting the certificate.

enrollment terminal

Example:

```
ciscoasa(config-ca-trustpoint)# enrollment terminal
ciscoasa(config-ca-trustpoint)#
```

Step 3 Import the certificate.

crypto ca authenticate *name*

Enter the name of the trustpoint you created for this certificate. Follow the prompts and paste the base-64 encoded certificate. Do not include the BEGIN CERTIFICATE and END CERTIFICATE lines.

```
ciscoasa(config-ca-trustpoint)# crypto ca authenticate ctx1
Enter the base 64 encoded CA certificate.
End with the word "quit" on a line by itself
```

Configure the Umbrella Connector Global Settings

The Umbrella global settings primarily define the API and Secret keys, or the legacy API token, that is needed to register the device with Cisco Umbrella.

The global settings are not sufficient to enable Umbrella. You must also enable Umbrella in your DNS inspection policy map, as described in Enable Umbrella in the DNS Inspection Policy Map, on page 9.

Before you begin

- Log into the Umbrella dashboard and generate an API key and a Secret key. Generate keys from **Admin** > **API Keys** in Umbrella. When generating the key, you must ensure that you copy and save the API and secret keys, as you cannot retrieve the secret later. The scope of the key must be read/write.
- (Legacy, not recommended.) Log into the Cisco Umbrella Network Devices Dashboard (https://login.umbrella.com/) and obtain a legacy network device API token for your organization. A token will be a hexadecimal string, for example, AABBA59A0BDE1485C912AFE. Generate a Legacy Network Devices API key from the Umbrella dashboard.
- Install the certificate for the Cisco Umbrella registration server.

Procedure

Step 1 Enter Umbrella configuration mode.

umbrella-global

Example:

```
ciscoasa(config)# umbrella-global
ciscoasa(config-umbrella)#
```

Step 2 Configure the API key and secret needed to register with Cisco Umbrella.

token-request-credential **api-key** *key_value* **secret** *secret_value*

Note that keys expire, and you need to disable umbrella, add new key/secret, and reenable umbrella when applying a new key.

Example:

```
ciscoasa(config)# umbrella-global
ciscoasa(config-umbrella)# token-request-credential
api-key f817feb474d94c56b3448bcbd08edc11 secret 4afe58df5c454161a10a172145cb1456
```

Step 3 (Legacy, not recommended.) Configure the API token needed to register with Cisco Umbrella.

You can configure the token only if you do not configure the API key and secret.

token *api_token*

Example:

```
ciscoasa(config)# umbrella-global
```

```
ciscoasa(config-umbrella)# token AABBA59A0BDE1485C912AFE
Please make sure all the Umbrella Connector prerequisites are satisfied:
1. DNS server is configured to resolve api.opendns.com
2. Route to api.opendns.com is configured
3. Root certificate of Umbrella registration is installed
4. Unit has a 3DES license
```

Step 4 (Optional.) If you intend to enable DNScrypt in the DNS inspection policy map, you can optionally configure the DNScrypt provider public key for certificate verification. If you do not configure the key, the default currently distributed public key is used for validation.

public-key *hex_key*

The key is a 32-byte hexadecimal value. Enter the hex value in ASCII with a colon separator for every 2 bytes. The key is 79 bytes long. Obtain this key from Cisco Umbrella.

The default key is:
B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79

To revert to using the default public key, enter **no public-key**. You can either omit the key you configured, or include it on the **no** version of the command.

Example:

```
ciscoasa(config-umbrella)# public-key
B735:1140:206F:225D:3E2B:D822:D7FD:691E:A1C3:3CC8:D666:8D0C:BE04:BFAB:CA43:FB79
```

Step 5 (Optional.) Configure the idle timeout after which a connection from a client to the Umbrella server will be removed if there is no response from the server.

timeout edns *hh:mm:ss*

The timeout is in hours:minutes:seconds format, and can be from 0:0:0 to 1193:0:0. The default is 0:02:00 (2 minutes).

Example:

```
ciscoasa(config-umbrella)# timeout edns 00:01:00
```

Step 6 (Optional.) Configure the local domain names for which Umbrella should be bypassed.

You can identify local domains for which DNS requests should bypass Cisco Umbrella and instead go directly to the configured DNS servers. For example, you can have your internal DNS server resolve all names for the organization's domain name on the assumption that all internal connections are allowed.

You can enter your local domain name directly. Optionally, you can create the regular expressions that define the name, then create a regular expression class map and specify it on the following command:

local-domain-bypass {*regular_expression* | **regex class** *regex_classmap*}

Example:

```
ciscoasa(config)# umbrella-global
ciscoasa(config-umbrella)# local-domain-bypass example.com
```

Step 7 (Optional.) Configure the addresses of the non-default Cisco Umbrella DNS servers, which resolve DNS requests, that you want to use.

resolver {**ipv4** | **ipv6**} *ip_address*

You can enter the command separately to define the IPv4 and IPv6 addresses of non-default Umbrella resolvers.

Example:

```
ciscoasa(config-umbrella)# resolver ipv4 208.67.222.222
ciscoasa(config-umbrella)# resolver ipv6 2620:119:35::35
```

Step 8 Select the method to use when registering with Umbrella

- **Token** (Legacy, not recommended.)—Enter the API token in the **Token** field.
- **Token-Request-Credential**—Configure the following:
  - **API-Key**—Enter the API key you generated from Umbrella.
  - **Secret-Key**—Enter the Secret key you were provided for the API key.

Enable Umbrella in the DNS Inspection Policy Map

Configuring the global Umbrella settings is not enough to register the device and enable DNS lookup redirection. You must add Umbrella as part of your active DNS inspection.

You can enable Umbrella globally by adding it to the preset_dns_map DNS inspection policy map.

However, if you have customized DNS inspection and applied different inspection policy maps to different traffic classes, you must enable Umbrella on each class where you want the service.

The following procedure explains how to implement Umbrella globally. If you have customized DNS policy maps, please see Configure DNS Inspection Policy Map.

Procedure

Step 1 Edit the preset_dns_map inspection policy map and enter parameter configuration mode.

```
ciscoasa(config)# policy-map type inspect dns preset_dns_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

Step 2 Enable Umbrella and optionally specify the name of the Cisco Umbrella policy to apply to the device.

umbrella [**tag** *umbrella_policy*] [**fail-open**]

The tag is the name of a policy as defined in Cisco Umbrella. During registration, Cisco Umbrella will assign the policy to the device (if the policy name exists). If you do not specify a policy, the default policy is applied.

Include the **fail-open** keyword if you want DNS resolution to work if the Umbrella DNS server is unavailable. When failing open, if the Cisco Umbrella DNS server is unavailable, Umbrella disables itself on this policy map and allows DNS requests to go to the other DNS servers configured on the system, if any. When the

Umbrella DNS servers are available again, the policy map resumes using them. If you do not include this option, DNS requests continue to go to the unreachable Umbrella resolver, so they will not get a response.

Example:

```
ciscoasa(config-pmap-p)# umbrella fail-open
```

Step 3 (Optional.) Enable DNScrypt to encrypt connections between the device and Cisco Umbrella.

dnscrypt

Enabling DNScrypt starts the key-exchange thread with the Umbrella resolver. The key-exchange thread performs the handshake with the resolver every hour and updates the device with a new secret key. Because DNScrypt uses UDP/443, you must ensure that the class map used for DNS inspection includes that port. Note that the default inspection class already includes UDP/443 for DNS inspection.

Example:

```
ciscoasa(config-pmap-p)# dnscrypt
```

Example

```
ciscoasa(config)# policy-map type inspect dns preset_dns_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# umbrella fail-open
ciscoasa(config-pmap-p)# dnscrypt
```

Verify the Umbrella Registration

After you configure the global Umbrella settings and enable Umbrella in DNS inspection, the device should contact Cisco Umbrella and register. You can check for successful registration by checking whether Cisco Umbrella provided a device ID.

First, check the service policy statistics, and look for the Umbrella Registration line. This should indicate the policy applied by Cisco Umbrella (the tag), the HTTP status of the connection , and the device ID.

The status should be 200 Success. Error codes indicate the following: 401 indicates that the API token was incorrect, 403 indicates that the scope of the key was not read/write (blocking access), 405 indicates that the API key has expired, and 409 indicates that the device already exists in Cisco Umbrella.

If the status is UNKNOWN, check for syslog messages. Message 339011: "Umbrella API token request received no responses," indicates that there is a routing problem for getting to api.umbrella.com, or you did not upload the required certificate.

Note that the Umbrella Resolver lines should not indicate that the resolvers are unresponsive. If they are, verify that you opened DNS communication to these IP addresses in your access control policy. This might be a temporary situation, or it might indicate a routing problem.

```
asa(config)# show service-policy inspect dns
Interface inside:
```

```
  Service-policy: global_policy
    Class-map: inspection_default
     Inspect: dns preset_dns_map, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate
0 pkts/sec, v6-fail-close 0 sctp-drop-override 0
       message-length maximum client auto, drop 0
       message-length maximum 512, drop 0
       dns-guard, count 0
       protocol-enforcement, drop 0
       nat-rewrite, count 0
      umbrella registration: mode: fail-open tag: default, status: 200 success, device-id:
010a13b8fbdfc9aa
         Umbrella ipv4 resolver: 208.67.220.220
         Umbrella ipv6 resolver: 2620:119:53::53
      Umbrella: bypass 0, req inject 0 - sent 0, res recv 0 - inject 0 local-domain-bypass
10
       DNScrypt egress: rcvd 402, encrypt 402, bypass 0, inject 402
       DNScrypt ingress: rcvd 804, decrypt 402, bypass 402, inject 402
       DNScrypt: Certificate Update: completion 10, failure 1
```

You can also verify the running configuration (filter on policy-map). The umbrella command in the policy map updates to show the device ID. You cannot directly configure the device ID when you enable this command. The following example edits the output to show the relevant information.

```
ciscoasa(config)# show running-config policy-map
!
policy-map type inspect dns preset_dns_map
parameters
  message-length maximum client auto
  message-length maximum 512
  dnscrypt
  umbrella device-id 010a3e5760fdd6d3
  no tcp-inspection
policy-map global_policy
class inspection_default
  inspect dns preset_dns_map
```

Examples for the Umbrella Connector

The following topics provide examples of configuring the Umbrella Connector.

Example: Enabling Umbrella on the Global DNS Inspection Policy

The following example shows how to enable Umbrella globally. The configuration enables DNScrypt using the default public key. It assigns the default Cisco Umbrella Enterprise Security policy. The key and secret are examples only; you must generate your own valid key/secret pair from Umbrella.

This example assumes that you have uploaded the appropriate certificate needed for the Umbrella registration. Ensure that you download the latest root certificate used by the Umbrella site, as the correct certificate changes over time.

```
ciscoasa(config)# dns domain-lookup outside
ciscoasa(config)# dns domain-lookup inside
ciscoasa(config)# dns name-server 208.67.220.220

ciscoasa(config)# umbrella-global
```

```
ciscoasa(config-umbrella)# token-request-credential
api-key f817feb474d94c56b3448bcbd08edc11 secret 4afe58df5c454161a10a172145cb1456

ciscoasa(config)# policy-map type inspect dns preset_dns_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# umbrella
ciscoasa(config-pmap-p)# dnscrypt
```

Example: Enabling Umbrella on an Interface with a Custom Inspection Policy

The following example shows how to enable Umbrella for a specific traffic class. Umbrella is enabled on the inside interface only for DNS/UDP traffic. Because we are enabling DNScrypt, UDP/443 must be included in the traffic class. An Enterprise Security policy named mypolicy (defined in Cisco Umbrella) is applied. The key and secret are examples only; you must generate your own valid key/secret pair from Umbrella.

This example assumes that you have uploaded the appropriate certificate needed for the Umbrella registration. Ensure that you download the latest root certificate used by the Umbrella site, as the correct certificate changes over time.

```
ciscoasa(config)# dns domain-lookup outside
ciscoasa(config)# dns domain-lookup inside
ciscoasa(config)# dns name-server 208.67.220.220

ciscoasa(config)# umbrella-global
ciscoasa(config-umbrella)# token-request-credential
api-key f817feb474d94c56b3448bcbd08edc11 secret 4afe58df5c454161a10a172145cb1456


ciscoasa(config)# policy-map type inspect dns umbrella-policy
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# umbrella tag mypolicy
ciscoasa(config-pmap-p)# dnscrypt

ciscoasa(config)# object-group service umbrella-service-object
ciscoasa(config-service-object-group)# service-object udp destination eq domain
ciscoasa(config-service-object-group)# service-object udp destination eq 443

ciscoasa(config)# access-list umbrella-acl extended permit
object-group umbrella-service-object any any

ciscoasa(config)# class-map dns-umbrella
ciscoasa(config-cmap)# match access-list umbrella-acl

ciscoasa(config)# policy-map inside-policy
ciscoasa(config-pmap)# class dns-umbrella
ciscoasa(config-pmap-c)# inspect dns umbrella-policy

ciscoasa(config)# service-policy inside-policy interface inside
```

Example: Excluding Certain Hosts or Networks from Umbrella Globally

If you need to exclude certain hosts or networks from using Umbrella, and the choice is global rather than interface-based, you can remove the global DNS inspection, and create separate classes for excluding or including Umbrella inspection.

The following example modifies the global inspection policy to exclude the 192.168.1.0/24 network from using Umbrella.

Before you begin

This example assumes that you have already enabled DNS and configured the Umbrella global settings.

Procedure

Step 1 Remove the global default DNS inspection.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# no inspect dns
```

Step 2 Create a DNS policy map that enables Umbrella.

In this example, the policy map is named umbrella-policy.

```
ciscoasa(config)# policy-map type inspect dns umbrella-policy
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# umbrella tag mypolicy
```

Step 3 Create a traffic class for the excluded traffic.

The following example uses an ACL to identify UDP/53 traffic from the 192.168.1.0/24 network.

```
ciscoasa(config)# access-list Umb_Exclude permit udp 192.168.1.0 255.255.255.0 any eq 53
ciscoasa(config)# class-map Umbrella_Exclude
ciscoasa(config-cmap)# match access-list Umb_Exclude
```

Step 4 Create a traffic class for hosts that should use Umbrella.

The following example matches UDP/53 traffic from any source.

```
ciscoasa(config)# class-map Umbrella_Include
ciscoasa(config-cmap)# match port udp eq 53
```

Step 5 Update the global inspection policy to enable DNS inspection for the traffic classes using the appropriate DNS policy map.

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)#  class Umbrella_Exclude
ciscoasa(config-pmap-c)# inspect dns
ciscoasa(config-pmap)#  class Umbrella_Include
ciscoasa(config-pmap-c)# inspect dns umbrella-policy
```

Monitoring the Umbrella Connector

The following topics explain how to monitor the Umbrella Connector.

Monitoring the Umbrella Service Policy Statistics

You can view both summarized and detailed statistics for DNS inspection with Umbrella enabled.

show service-policy inspect dns [**detail**]

Without the **detail** keyword, you see all the basic DNS inspection counters plus Umbrella configuration information. The status field provides the HTTP status code for the system's attempt to register with Cisco Umbrella.

The Resolver lines indicate which Umbrella servers are being used. These lines will say whether the server is **unresponsive**, or if the system is currently **probing** the server to determine if it has become available. If the mode is fail-open, the system allows DNS requests to go to other DNS servers (if configured); otherwise, DNS requests will not get a response so long as the Umbrella servers are unreponsive.

```
asa(config)# show service-policy inspect dns
Interface inside:
  Service-policy: global_policy
    Class-map: inspection_default
     Inspect: dns preset_dns_map, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate
 0 pkts/sec, v6-fail-close 0 sctp-drop-override 0
        message-length maximum client auto, drop 0
        message-length maximum 512, drop 0
        dns-guard, count 0
        protocol-enforcement, drop 0
        nat-rewrite, count 0
       umbrella registration: mode: fail-open tag: default, status: 200 success, device-id:
 010a13b8fbdfc9aa
          Umbrella ipv4 resolver: 208.67.220.220
          Umbrella ipv6 resolver: 2620:119:53::53
       Umbrella: bypass 0, req inject 0 - sent 0, res recv 0 - inject 0 local-domain-bypass
 10
        DNScrypt egress: rcvd 402, encrypt 402, bypass 0, inject 402
        DNScrypt ingress: rcvd 804, decrypt 402, bypass 402, inject 402
        DNScrypt: Certificate Update: completion 10, failure 1
```

The detailed output shows DNScrypt statistics and the keys used.

```
asa(config)# show service-policy inspect dns detail
Global policy:
  Service-policy: global_policy
    Class-map: inspection_default
    Class-map: dnscrypt30000
      Inspect: dns dns_umbrella, packet 12, lock fail 0, drop 0, reset-drop 0,
               5-min-pkt-rate 0 pkts/sec, v6-fail-close 0 sctp-drop-override 0
        message-length maximum client auto, drop 0
        message-length maximum 1500, drop 0
        dns-guard, count 3
        protocol-enforcement, drop 0
        nat-rewrite, count 0
       Umbrella registration: mode: fail-open tag: default, status: 200 SUCCESS, device-id:
 010af97abf89abc3, retry 0
          Umbrella ipv4 resolver: 208.67.220.220
          Umbrella ipv6 resolver: 2620:119:53::53
       Umbrella: bypass 0, req inject 6 - sent 6, res recv 6 - inject 6 local-domain-bypass
 10
          Umbrella app-id fail, count 0
          Umbrella flow alloc fail, count 0
          Umbrella block alloc fail, count 0
          Umbrella client flow expired, count 0
```

```
        Umbrella server flow expired, count 0
        Umbrella request drop, count 0
        Umbrella response drop, count 0
      DNScrypt egress: rcvd 6, encrypt 6, bypass 0, inject 6
      DNScrypt ingress: rcvd 18, decrypt 6, bypass 12, inject 6
        DNScrypt length error, count 0
        DNScrypt add padding error, count 0
        DNScrypt encryption error, count 0
        DNScrypt magic_mismatch error, count 0
        DNScrypt disabled, count 0
        DNScrypt flow error, count 0
        DNScrypt nonce error, count 0
      DNScrypt: Certificate Update: completion 1, failure 1
        DNScrypt Receive internal drop count 0
        DNScrypt Receive on wrong channel drop count 0
        DNScrypt Receive cannot queue drop count 0
        DNScrypt No memory to create channel count 0
        DNScrypt Send no output interface count 1
        DNScrypt Send open channel failed count 0
        DNScrypt Send no handle count 0
        DNScrypt Send dupb failure count 0
        DNScrypt Create cert update no memory count 0
        DNScrypt Store cert no memory count 0
        DNScrypt Certificate invalid length count 0
        DNScrypt Certificate invalid magic count 0
        DNScrypt Certificate invalid major version count 0
        DNScrypt Certificate invalid minor version count 0
        DNScrypt Certificate invalid signature count 0
        Last Successful: 01:42:29 UTC May 2 2018, Last Failed: None
        Magic DNSC, Major Version 0x0001, Minor Version 0x0000,
        Query Magic 0x714e7a696d657555, Serial Number 1517943461,
        Start Time 1517943461 (18:57:41 UTC Feb 6 2018)
        End Time 1549479461 (18:57:41 UTC Feb 6 2019)
        Server Public Key
240B:11B7:AD02:FAC0:6285:1E88:6EAA:44E7:AE5B:AD2F:921F:9577:514D:E226:D552:6836
        Client Secret Key Hash
48DD:E6D3:C058:D063:1098:C6B4:BA6F:D8A7:F0F8:0754:40B0:AFB3:CB31:2B22:A7A4:9CEE
        Client Public key
6CB9:FA4B:4273:E10A:8A67:BA66:76A3:BFF5:2FB9:5004:CD3B:B3F2:86C1:A7EC:A0B6:1A58
        NM key Hash
9182:9F42:6C01:003C:9939:7741:1734:D199:22DF:511E:E8C9:206B:D0A3:8181:CE57:8020
```

Monitoring Umbrella Syslog Messages

You can monitor the following Umbrella-related syslog messages:

- %ASA-3-339001: DNSCRYPT certificate update failed for *number* tries.

  Check that there is a route to the Umbrella server and that the egress interface is up and functioning correctly. Also check that the public key configured for DNScrypt is correct. You might need to obtain a new key from Cisco Umbrella.

- %ASA-3-339002: Umbrella device registration failed with error code *error_code*.

  The error codes have the following meanings:

  - 400—There is a problem with the request format or content. The token is probably too short or corrupted. Verify that the token matches the one on the Umbrella Dashboard.
  - 401—The API token is not authorized. Try reconfiguring the token. If you refreshed the token on the Umbrella Dashboard, then you must ensure that you use the new token.

  - 409—The device ID conflicts with another organization. Please check with the Umbrella Administrator to see what the issue might be.
  - 500—There is an internal server error. Check with the Umbrella Administrator to see what the issue might be.

- %ASA-6-339003: Umbrella device registration was successful.
- %ASA-3-339004: Umbrella device registration failed due to missing token.

  You must obtain an API token from Cisco Umbrella and configure it in the global Umbrella settings.
- %ASA-3-339005: Umbrella device registration failed after *number* retries.

  Check the syslog 339002 messages to identify the errors that you need to fix.
- %ASA-3-339006: Umbrella resolver *IP_address* is reachable, resuming Umbrella redirect.

  This message indicates that the system is functioning normally again. No action is needed.
- %ASA-3-339007: Umbrella resolver *IP_address* is unresponsive and fail-close mode used, starting probe to resolver.

  Because you are using fail-close mode, users will not get responses to their DNS requests until the Umbrella DNS server comes back online. If the problem persists, verify that there is a route from the system to the Umbrella servers, and that you allow DNS traffic to the servers in your access control policy.
- Messages related to API key/secret registration:
  - %ASA-6-339010: Umbrella API token request was successful
  - %ASA-3-339011: Umbrella API token request received no responses
  - %ASA-3-339012: Umbrella API token request failed with error code %d
  - %ASA-3-339013: Umbrella API token request failed in response processing
  - %ASA-3-339014: Umbrella API token request failed after %d retries; abort registration

History for Cisco Umbrella Connector

Feature Name	Platform Releases	Description
Cisco Umbrella support.	9.10(1)	You can configure the device to redirect DNS requests to Cisco Umbrella, so that your Enterprise Security policy defined in Cisco Umbrella can be applied to user connections. You can allow or block connections based on FQDN, or for suspicious FQDNs, you can redirect the user to the Cisco Umbrella intelligent proxy, which can perform URL filtering. The Umbrella configuration is part of the DNS inspection policy. We added or modified the following commands: **umbrella**, **umbrella-global**, **token**, **public-key**, **timeout edns**, **dnscrypt**, **show service-policy inspect dns detail**.

Feature Name	Platform Releases	Description
Cisco Umbrella Enhancements.	9.12(1)	You can now identify local domain names that should bypass Cisco Umbrella. DNS requests for these domains go directly to the DNS servers without Umbrella processing. You can also identify which Umbrella servers to use for resolving DNS requests. Finally, you can define the Umbrella inspection policy to fail open, so that DNS requests are not blocked if the Umbrella server is unavailable. We added or changed the following commands: **local-domain-bypass**, **resolver**, **umbrella fail-open**.
New Umbrella API.	9.23(1)	You can now configure Umbrella using the Umbrella Open API, which uses an API key with a Secret key. We added the following command: **token-request-credential**