Application Visibility and Control

The following topics explain how to enable and configure Application Visibility and Control (AVC). AVC makes it possible for you to write access control rules based on applications rather than just IP addresses and ports. You can also use AVC with any feature that uses extended access control lists (ACL).

About Application Visibility and Control

You can use access control rules to filter traffic based on the application used in the connection. This is called Application Visibility and Control. The system can recognize a wide variety of applications, so that you do not need to figure out how to block one web application without blocking all web applications.

The Vulnerability Database and Network-Service Objects

When you enable Application Visibility and Control, the system downloads the Vulnerability Database (VDB), the same one used by Threat Defense devices. Once downloaded, the system automatically creates the following:

- Network-service objects for each application. The VDB typically includes more than 4000 defined applications. Each application includes a name, application category, description, app ID, and a list of associated domain names.
- Network-service object groups for each application category. The VDB typically includes more than 60 defined categories.

The AVC objects are created as dynamic network-service objects or groups, so they are not saved to the running configuration. You can also create your own categories by creating custom network-service groups that incorporate dynamic VDB-created objects.

You can then use the AVC or custom network-service groups in access control rules or other extended ACLs.

The system downloads an updated VDB weekly. You can also force a download at any time. Downloads update the existing objects. Any access control rules or other ACLs that use these objects automatically use the updated objects.

Determining the Application in a Connection

Operationally, the system uses the following techniques to determine the application used in a connection based on domain name, so that the connection can be correctly matched to an AVC-based access control entry. The system uses these techniques to build the IP-to-domain mapping cache needed to match an IP address to an application. In a high-availability group, the cache is replicated on the standby unit.

- DNS request/response snooping—The DNS request/response queries must be on port UDP/53 and not encrypted. They must go through the device; if DNS resolution happens on a path that bypasses the ASA, then DNS responses cannot be snooped. If the destination IP address/protocol/port in a connection is in the DNS IP-to-domain cache, it can be matched to the right AVC-based access control entry in the first packet of the connection.
- TLS Client Hello snooping for HTTPS connections—For HTTPS requests, the optional SNI field contains the domain name for the requested server. For snooping to obtain this name, the SNI field must be present and not encrypted. Snooping extracts the domain name and associates it with the IP address of the TLS Client Hello packet and adds it to the DNS snooping cache. Note that the SNI field is optional in the client hello, so it might not be available to snoop. When using this type of snooping, application identification is not yet available on the first packet of the first connection to the destination server, but subsequent connections should match the correct AVC-based access rule.
- HTTP request header hostname snooping—The HTTP Host header includes the domain name. Snooping extracts the domain name and associates it with the IP address of the destination of the HTTP packet. When using this type of snooping, application identification is not yet available on the first packet of the first connection to the destination server, but subsequent connections should match the correct AVC-based access rule.

If these items are not available, then AVC classification is not possible and hit counts for all applications will be zero. In this case, your AVC rules will not function correctly.

Note In a Content Delivery Network (CDN), an IP address might be mapped to multiple domains. This can result in the misclassification of application traffic, and therefore a connection might match the wrong access control rule and be inadvertently blocked or allowed. You can view the DNS snooping cache to determine if an IP address matches more than one domain. See Monitoring the DNS Snooping Cache, on page 17.

Features that Support AVC

You can use AVC network-service object groups in any policy that supports extended access control lists (ACL). This includes access control rules, QoS policing service policy rules, other service policy rules, route maps, and so forth.

What Users See on Blocked Connections

When a user tries to access a domain name that you are blocking with deny access control rules, the browser shows a generic Unable to Connect error page. This message might suggest retrying the connection. You might want to set user expectations about allowed and disallowed applications on your network to avoid service calls.

If the user is using an application for the connection, the error seen depends on the application behavior. You cannot customize the error page or message displayed to the user.

AVC-based deny rules on extended ACLs used for services other than access control simply omit those connections from the service. These actions should be transparent to end users.

Licensing for Application Visibility and Control

Application Visibility and Control requires the following licenses:

- Carrier
- Strong Encryption (3DES). Your Smart License account must allow export-controlled functionality.

Prerequisites for Application Visibility and Control

Model Requirements

- Secure Firewall 6100

Guidelines and Limitations for Application Visibility and Control

Firewall Mode Guidelines

In multiple context mode, the VDB and related files are shared by all contexts. However, AVC network-service objects and groups and other data structures are created in each context. You enable AVC per user context.

VDB Download Guidelines

- You must configure DNS and DNS lookup on an interface through which support.sourcefire.com can be contacted, so that the Vulnerability Database (VDB) can be downloaded. A route to that server must also be available.
- In a high-availability group, the VDB download must be possible over the management interface, because the standby unit cannot pass traffic over data interfaces. The standby unit downloads VDB updates itself.
- In a cluster, each unit downloads the VDB separately.
- In multiple context mode, only one user context will download the VDB, and share it with contexts where AVC is enabled.

- VDB download uses HTTPS, so it requires that a trusted CA certificate from third-party Certificate Authority is installed on the device, to ensure a secure connection.
- When downloading a VDB update, after the download is complete, the system clears the existing AVC network-service objects and groups, and other AVC files, and rebuilds them. Your AVC-based policies will stop working until the update is complete.
- The VDB download file is about 70 MB. An extracted VDB requires at least 450 MB of disk space. Ensure there is adequate free space on the device to download and extract the VDB.

Additional Guidelines

- For AVC traffic classification based on DNS snooping to work, you must enable DNS inspection (which is enabled by default) and configure a DNS trusted source (**dns trusted-source** command).
- Do not enable DNScrypt. If DNScrypt is enabled, no DNS snooping is possible, because it encrypts the DNS connection. Check your DNS inspection policy to ensure the **dnscrypt** command is not enabled.
- DNS snooping supports DNS requests over UDP/53 only. DNS snooping does not work with DNS requests over TCP/53 or HTTP/HTTPS.
- For TLS Client Hello and HTTP header snooping, traffic classification is not completed with the first packet for a newly seen application. However, subsequent connections that use the application should be matched to the expected rule.,
- QUIC and DTLS connections cannot be snooped for IP-to-domain mapping information.

Configure Application Visibility and Control

Application Visibility and Control (AVC) requires some initial setup, but once the feature is enabled, you can create access control rules and extended ACLs using network-service object groups as usual. The following procedure covers all aspects of enabling and using AVC.

Procedure

Step 1 Configure the Carrier license and ensure that your Smart License account is enabled for export-controlled features.

For more information, see the licensing chapters in the the appropriate release of the ASA General Operations CLI Configuration Guide.

Step 2 Configure DNS servers and DNS lookup on the interfaces.

Add DNS servers to the DefaultDNS group (or a custom group) and enable DNS lookup on each interface where you want to discover application usage.

For more information, see the information on configuring the DNS servers in the Basic Settings chapter in the appropriate release of the ASA General Operations CLI Configuration Guide.

Step 3 Configure trusted DNS servers for DNS snooping.

When DNS names are included in network-service objects, the system snoops DNS request/response traffic to gather IP addresses for DNS domain names and caches the results. Any DNS request/response can be snooped.

For security reasons, you can limit the scope of DNS snooping by defining which DNS servers should be trusted. Any DNS traffic to non-trusted DNS servers is ignored and not used to obtain mappings for network-service objects.

By default, all configured and learned DNS servers are trusted; you need to change this only if you want to limit the trusted list. Use the **show dns** command to determine the currently DNS Trusted Source configuration.

Specifically, you must ensure that the DNS servers your clients use to resolve domain names are on the trusted list. The default configuration might handle the situation where user DNS servers are configured through DHCP. However, if you configure client DNS servers explicitly, ensure that the IP addresses of those servers are on the trusted list.

For more information, see Configure Trusted DNS Servers.

Step 4 Configure DNS inspection.

DNS inspection is enabled by default. If you have turned it off, you need to enable it again. The default DNS inspection global configuration is adequate for AVC. For more information about DNS inspection, see DNS Inspection.

Step 5 Configure a route to support.sourcefire.com.

Ensure that either a static route, or updates from a routing protocol, include a path to the server used for downloading the Vulnerability Database (VDB). For HA units, this route must be accessible from the management interface. If you are unsure, use **ping** to verify that a route exists.

For information about configuring static routes and routing protocols, see the appropriate release of the ASA General Operations CLI Configuration Guide.

Note
This feature does not work on an air-gapped network. AVC is meaningful only for a network that can access the internet.

Step 6 Configure a valid device identity certificate from a third-party CA on the device. The certificate is needed to validate the connection between the device and the VDB download server.

Step 7 Enable Application Visibility and Control, on page 6

Step 8 Create Custom Application Categories, on page 7

Step 9 Configure AVC Access Rules, on page 8

Step 10 (Optional.) Use AVC objects in other features where you need application control.

You can use AVC-based network-service object groups in any extended ACLs. You can then use those ACLs in any feature that uses an extended ACL, such as service policy class maps, route maps, and so forth.

The procedure explained in Configure AVC Access Rules, on page 8 generally applies to adding AVC-based access control entries to extended ACLs.

Step 11 (Optional.) Manually Download the VDB, on page 10

Enable Application Visibility and Control

You must enable Application Visibility and Control (AVC) to download the Vulnerability Database (VDB) and create the network-service objects and groups that you can use in access control rules and extended ACLs.

Note If you subsequently disable AVC, all downloaded and extracted files, and the AVC network-service objects and groups, are deleted. Use the **no avc** command to disable AVC.

Before you begin

Ensure that you have met the requirements listed in Guidelines and Limitations for Application Visibility and Control, on page 3. Otherwise, the VDB download might fail.

Procedure

Step 1 Enable Application Visibility and Control.

avc

Note
When you view the running configuration, the **avc** command shows the version of the VDB currently installed. This parameter is not configurable.

Example:

```
ciscoasa(config)# avc
```

Step 2 Wait for the VDB to be downloaded and extracted, and the network-service objects and groups to be created.

Use the **show avc status** command to view the status of the AVC system. For example, the following output shows that the system is ready, and the VDB is the most current version.

```
ciscoasa(config)# show avc status
AVC (Application Visibility and Control): ENABLED : READY
VDB (containing app definition) download status: UP-TO-DATE
VDB version: current 397, last 0
VDB last update at 15:29:57 UTC Apr 29 2025; last update attempt at 15:27:36 UTC Apr 29
2025; next update at 15:30:55 UTC May 6 2025.
VDB download link:
https://50.19.123.95/auto-update/auto-dl.cgi/Download/files/Cisco_VDB_Fingerprint_Database-4.5.0-399.sh.REL.tar
```

You can also use syslog messages to track status. See Monitoring AVC Syslog Messages, on page 18.

Create Custom Application Categories

You can create new network-service object groups that incorporate the application dynamic network-service objects created from the VDB download. Use them as you would any other AVC network-service object group.

Creating a custom application category is useful if you want to allow or block a subset of a pre-defined AVC category. For example, if you want to block most of the applications in the _gaming_ AVC category, but allow a few of the applications that are in the category, you would create a custom category of the allowable applications. You could then write an access control rule that allowed your custom category and follow it with a rule that disallows the AVC _gaming_ category.

Because access control rules and extended ACLs allow the use of network-service groups only, not network-service objects, you must create custom categories if you want to write rules that do not conform to the existing AVC categories.

Procedure

Step 1 Collect the list of network-service object names for the applications you want to treat in a like manner.

The following are some methods you might find useful:

- **show avc top** *n*

  The application names are the object names. For example, **show avc top 20** lists the 20 most seen applications by the device. This method provides you with the names of applications that are actively in use in the network, so that access rules that address these applications can have an impact on network traffic.

- **show object-group network-service _avc_visibility_nsg_ detail**

  show object-group network-service _avc_visibility_nsg_ detail | include hitcnt=[1-9]

  This display includes every AVC application along with their hit count. The output is long but provides a complete list of AVC network-service object names. You can add the filter to see only those applications that have non-zero hit counts.

- **show object network-service detail**

  show object network-service detail | include hitcnt=[1-9]

  Shows all network-service objects, including the AVC applications. The output is long but complete. You can add the filter to see only those applications that have non-zero hit counts.

Step 2 Create the network-service object group for your application category.

a) Create the network-service object group.

object-group network-service *name*

Where *name* is the name of your category.

b) Add network-service objects (applications) to the group.

network-service-member *name*

For example, the following commands create a custom application category named my_social_networking. You can then use the my_social_networking network-service object group in an access control rule or another extended ACL entry.

```
object-group network-service my_social_networking
  network-service-member "TikTok"
  network-service-member "Facebook"
   network-service-member "WhatsApp"
```

Configure AVC Access Rules

Use the AVC network-service dynamic object groups, and any custom network-service object groups you create, to configure allow and block rules in your access control policy. As VDB downloads change the dynamic objects and groups, the access control rules automatically pick up the changes and apply them to matching connections.

To determine rule matching for an application, the system uses the IP address of the connection's source and destination to look up the related domain names in the IP address-to-domain cache. Note that a single IP address might be mapped to multiple domains.

Using this information, along with protocol/port, the system determines which network-service groups contain the application. Then, the access control rules (or generic extended ACL entries for features other than access control) are searched for those groups, the first match wins. Thus, if an application appears in more than one category, ensure that your rule set applies the intended action to that application.

Before you begin

Either identify the AVC network-service group you want to use in the rule or create the custom network-service object groups you need.

This procedure assumes that you have global access control rules or interface-based access control rules already defined, and you are adding rules to existing access groups. If you are instead starting from scratch, see Access Rules.

Procedure

Add the AVC-based access control rule to the appropriate ACL.

An AVC-based access control rule will normally use a network-service object group as the destination address. This controls internal users' access to an application.

Use the following command to add an access control list entry.

access-list *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} *protocol_argument source_address_argument dest_address_argument* [**log** [[*level*] [**interval** *secs*] | **disable** | **default**]] [**time-range** *time_range_name*] [**inactive**]

The options are:

- *access_list_name*—The name of the new or existing ACL.

- Line number—The **line** *line_number* option specifies the line number at which to insert the ACE; otherwise, the ACE is added to the end of the ACL.
- Permit or Deny—The **deny** keyword denies or exempts a packet if the conditions are matched. The **permit** keyword permits or includes a packet if the conditions are matched.
- Protocol—The *protocol_argument* specifies the IP protocol. You can limit the protocol or specify **ip**. Because network-service objects can contain port/protocol, we recommend using **ip** to avoid potential problems. If there is a mismatch between this argument and the contents of the object, the rule will never be matched by any connections.
- Source Address, Destination Address—The *source_address_argument* specifies the IP address or FQDN from which the packet is being sent, and the *dest_address_argument* specifies the IP address or FQDN to which the packet is being sent. To create an AVC rule, at least one of these must use the **object-group-network-service** argument. Typically, an AVC rule will use **object-group-network-service** for the destination address.
  - **host** *ip_address*—Specifies an IPv4 host address.
  - *ip_address mask*—Specifies an IPv4 network address and subnet mask, such as 10.100.10.0 255.255.255.0.
  - *ipv6-address/prefix-length*—Specifies an IPv6 host or network address and prefix.
  - **any**, **any4**, and **any6**—**any** specifies both IPv4 and IPv6 traffic; **any4** specifies IPv4 traffic only; and **any6** specifies IPv6 traffic only.
  - **interface** *interface_name*—Specifies the name of an ASA interface. Use the interface name rather than IP address to match traffic based on which interface is the source or destination of the traffic.
  - **object** *nw_obj_id*—Specifies a network object created using the **object network** command.
  - **object-group** *nw_grp_id*—Specifies a network object group created using the **object-group network** command.
  - **object-group-network-service** *name*—Specifies the name of a network-service object group.
- Logging—**log** arguments set logging options when an ACE matches a connection for network access (an ACL applied with the **access-group** command). If you enter the **log** option without any arguments, you enable syslog message 106100 at the default level (6) and for the default interval (300 seconds). Log options are:
  - *level*—A severity level between 0 and 7. The default is 6 (informational). If you change this level for an active ACE, the new level applies to new connections; existing connections continue to be logged at the previous level.
  - **interval** *secs*—The time interval in seconds between syslog messages, from 1 to 600. The default is 300. This value is also used as the timeout value for deleting an inactive flow from the cache used to collect drop statistics.
  - **disable**—Disables all ACE logging.
  - **default**—Enables logging to message 106023 for denied packets. This setting is the same as not including the **log** option.

- Time Range—The **time-range** *time_range_name* option specifies a time range object, which determines the times of day and days of the week in which the ACE is active. If you do not include a time range, the ACE is always active.
- Activation—Use the **inactive** option to disable the ACE without deleting it. To reenable it, enter the entire ACE without the inactive keyword.

Example:

For example:

```
ciscoasa(config)# access-list dmz permit ip any
object-group-network-service _business_
```

Example

The following example allows business applications on the dmz interface, but blocks gaming applications.

```
access-list dmz permit ip any object-group-network-service _business_
access-list dmz deny ip any object-group-network-service _gaming_
```

Manually Download the VDB

The Vulnerability Database (VDB) is automatically downloaded when you enable AVC. It is then automatically refreshed weekly. However, if you need to force an update, you can manually download the latest VDB.

Before you begin

Ensure that you have met the requirements listed in Guidelines and Limitations for Application Visibility and Control, on page 3. Otherwise, the VDB download might fail.

Procedure

Step 1 Start the VDB download.

avc download vdb

Example:

```
ciscoasa(config)# avc download vdb
```

Step 2 Wait for the VDB to be downloaded and extracted, and the network-service objects and groups to be created.

Use the **show avc status** command to view the status of the AVC system. For example, the following output shows that the system is ready, and the VDB is the most current version.

```
ciscoasa(config)# show avc status
```

```
AVC (Application Visibility and Control): ENABLED : READY
VDB (containing app definition) download status: UP-TO-DATE
VDB version: current 397, last 0
VDB last update at 15:29:57 UTC Apr 29 2025;
last update attempt at 15:27:36 UTC Apr 29 2025; next update at 15:30:55 UTC May 6 2025.
VDB download link: https://50.19.123.95/auto-update/auto-dl.cgi/Download/
files/Cisco_VDB_Fingerprint_Database-4.5.0-397.sh.REL.tar
```

You can also use syslog messages to track status. See Monitoring AVC Syslog Messages, on page 18.

Monitor and Troubleshoot Application Visibility and Control

The following topics explain how you can monitor and troubleshoot AVC.

Monitoring and Troubleshooting AVC and VDB Download Status

You can view the status of the AVC system, including VDB information, using the following command:

show avc status

For example:

```
AVC (Application Visibility and Control): ENABLED : READY
VDB (containing app definition) download status: UP-TO-DATE
VDB version: current 397, last 0
VDB last update at 15:29:57 UTC Apr 29 2025; last update attempt
  at 15:27:36 UTC Apr 29 2025; next update at 15:30:55 UTC May 6 2025.
VDB download link: https://50.19.123.95/auto-update/auto-dl.cgi/
Download/files/Cisco_VDB_Fingerprint_Database-4.5.0-397.sh.REL.tar
```

The possible AVC statuses are:

- ENABLED/DISABLED—Whether the AVC feature is turned on.
- READY—The Vulnerability Database (VDB) download was successful, the creation of applications and categories was successful, and you can now use AVC.
- NOT READY—There was a problem with the VDB download or the creation of applications and categories. Check VDB status.

The possible VDB download statuses are:

- UP-TO-DATE. The VDB download was successful and the most recent version is installed.
- INITIALIZATION. The download is starting.
- PROGRESSING. The download is in progress.
- RETRY. The download failed and a retry attempt is in progress.
- FAILED. The system has made 6 attempts to download the VDB but was not able to perform the download. Check syslog messages for an indication of the problem, such as 861003. The main reasons for failure are:

- The VDB server domain is not resolving. Check your DNS configuration and ensure that the interface through which updates are attempted is enabled for domain lookup. Try the copy command to see if it fails in a way that indicates the domain name is not resolving:

```
ciscoasa(config)# copy https://support.sourcefire.com/index.html index.html
Address or name of remote host [support.sourcefire.com]?
?Invalid host address or name
%Error parsing filename (No such device)
```

- The VDB server is unreachable. Check your routing configuration and install a static route if necessary. Verify that the interface for the route is up and that pinging the download host works.
- SSL validation is failing. Ensure you install a third-party CA certificate that can be used to validate the connection. If downloads used to work, check whether the certificate used has expired.
- The device license does not support strong encryption. In this case, you cannot use AVC.

Monitoring Application Usage

Even if you are not using applications to control access, you can monitor which applications are being used in the network.

The following commands can provide hit count information on the applications seen in the connections handled by the device.

- **show avc top** *n*

  The list shows the *n* number of most-hit applications based on connections. The following example shows the top 3 applications:

```
ciscoasa# show avc top 3
AVC: Top 3 App hits
  Application: TikTok, Hit Count: 33950
  Application: GameSpot, Hit Count: 14400
  Application: Facebook, Hit Count: 980
```

- **show object-group network-service _avc_visibility_nsg_ detail**

  show object-group network-service _avc_visibility_nsg_ detail | include hitcnt=[1-9]

  The _avc_visibility_nsg_ is created from the VDB and contains all AVC applications. By showing the content of this object, you can view hit counts for all AVC-defined applications. The output is long but provides a complete list of AVC network-service object names. You can add the filter to see only those applications that have non-zero hit counts. The following is a partial output:

```
ciscoasa# show object-group network-service _avc_visibility_nsg_ detail
object-group network-service _avc_visibility_nsg_ dynamic (id=0xfdff0000) (hitcnt=1391)

 network-service-member "ADrive" dynamic (hitcnt = 0)
  description Online file storage and backup.
  domain adrive.com (bid=149781) ip (hitcnt=0)
 network-service-member "Amazon" dynamic (hitcnt = 12)
  description Online retailer of books and most other goods.
  domain amazon.com (bid=353725) ip (hitcnt=12)
  domain amazon.jobs (bid=399283) ip (hitcnt=0)
  domain amazon.in (bid=557827) ip (hitcnt=0)
```

```
  domain amazon.es (bid=713345) ip (hitcnt=0)
  domain amazon.de (bid=893113) ip (hitcnt=0)
  domain amazon.co.uk (bid=1047807) ip (hitcnt=0)
  domain amazon.co.jp (bid=1068127) ip (hitcnt=0)
  domain amazon.ca (bid=1287853) ip (hitcnt=0)
  domain m.media-amazon.com (bid=1347445) ip (hitcnt=0)
```

- **show object network-service detail**

  show object network-service detail | include hitcnt=[1-9]

  The output shows all network-service objects, including any you have created. Objects that include the keyword **dynamic** and the **app-id** command are those created by AVC. You can include an object name in the command to limit the output. Note that AVC creates 4000+ network-service objects, so the output will be extensive. You can add the filter to see only those applications that have non-zero hit counts.

```
ciscoasa# show object network-service detail
object network-service "ADrive" dynamic (hitcnt = 0)
 description Online file storage and backup.
 app-id 17
 domain adrive.com (bid=2130077667) ip (hitcnt=0)
object network-service "Amazon" dynamic (hitcnt = 0)
 description Online retailer of books and most other goods.
 app-id 24
 domain amazon.com (bid=2130286049) ip (hitcnt=0)
 domain amazon.jobs (bid=2130395289) ip (hitcnt=0)
 domain amazon.in (bid=2130530143) ip (hitcnt=0)
 domain amazon.es (bid=2130681663) ip (hitcnt=0)
 domain amazon.de (bid=2130718733) ip (hitcnt=0)
 ..
```

- **show service policy**

  If you use network-service object groups in an extended ACL that you use for matching criteria in a service policy rule, this command shows the contents of those object groups and their hit counts.

  For example, the following output shows hit counts for a policing policy.

```
ciscoasa# show service-policy police

Interface inside:
  Service-policy: throttle_map
    Class-map: app_throttled
      match access-list app_throttled
        network service object group/category _mobile_application_
          WeChat (hitcnt=4) WhatsApp (hitcnt=2)
      Input police Interface inside:
        cir 8000 bps, bc 1500 bytes
        conformed 59 packets, 5781 bytes; actions:  transmit
        exceeded 150 packets, 14700 bytes; actions:  drop
        conformed 0 bps, exceed 16 bps
    Class-map: custom_app_throttled
      match access-list custom_app_throttled
        network service object group/category custom_app_throttled
          Twitter/X (hitcnt=2)
      Input police Interface inside:
        cir 16000 bps, bc 1500 bytes
        conformed 70 packets, 6860 bytes; actions:  transmit
        exceeded 29 packets, 2842 bytes; actions:  drop
        conformed 8 bps, exceed 0 bps
```

Troubleshooting Application Classification

After you enable AVC, the VDB is successfully downloaded, and the AVC network-service objects and groups are created, the system starts snooping and building the IP address-to-domain name cache. As connections containing addresses or domains that appear in the cache are attempted/established, hit counts for the applications increment. Thus, each connection is classified as belonging to a particular application based on the IP address to domain name mapping.

Give the device some time to snoop traffic, build a cache, and collect hit count information. You can then examine the hit counts to get a sense of which applications are being used on the network segment controlled by the device.

The following topics address the primary problems you might encounter with application classification.

Verify that Application Classification is Happening

For AVC to work at all, applications must be classified. That is, the DNS snooping cache must be built to map IP addresses to domain names.

Use the **show avc top** *n* command to verify that some applications have hit counts. For example, you could show the top 3 applications, as follows.

```
ciscoasa: show avc top 3
AVC: Top 3 App hits
  Application: TikTok, Hit Count: 33950
  Application: GameSpot, Hit Count: 14400
  Application: Facebook, Hit Count: 980
```

If no applications have hit counts, then nothing is being classified, and any AVC-based rules will be inoperable. For example, the following indicates no applications are being classified:

```
ciscoasa# show avc top

AVC: Top 20 App hits
  Application: ADrive, Hit Count: 0
  Application: Amazon, Hit Count: 0
  ...
  ...
  Application: Dropbox, Hit Count: 0
  Application: eBay, Hit Count: 0
  Application: eBay Bid, Hit Count: 0
```

You can also check the DNS snooping cache to see if it is empty.

```
ciscoasa# show dns ip-cache
DNS snooping IP cache: 0 in use, 0 most used
Address          Domain                   Idle(sec) Timeout   Hit-count  Source
```

Do the following to troubleshoot a zero-hit-count situation.

Procedure

Step 1 Verify that DNS inspection is enabled and that it has a non-zero packet count.

```
ciscoasa# show service-policy inspect dns
  Global policy:
   Service-policy: global_policy
    Class-map: inspection_default
      Inspect: dns migrated_dns_map_1, 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
```

If **inspect dns** is present, but the packet count is zero, it means that no DNS request/response traffic on UDP/53 is going through the device, and hence there are DNS queries to snoop.

If DNS inspection is not enabled, enable it now, give the system time to snoop DNS, and try the **show avc top** command again.

Step 2 If DNS inspection is enabled and the packet count is not 0, verify that DSNcrypt is not enabled.

If the **show service-policy inspect dns** output includes DNScrypt lines such as the following, DNScrypt is enabled. Remove the **dnscrypt** command from the **inspect dns** configuration.

```
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
```

Step 3 If DNS inspection is enabled and the packet count is not 0, verify that trusted DNS servers include the servers used by your clients. If trusted servers list does not include the servers used by clients, add them now.

The **show dns** command provides the current view of DNS trust. The following output indicates DNS trust is configured with default settings.

```
ciscoasa# show dns
INFO: no activated FQDN
DNS Trusted Source enabled for DHCP Server Configured
DNS Trusted Source enabled for DHCP Client Learned
DNS Trusted Source enabled for DHCP Relay Learned
DNS Trusted Source enabled for DNS Server Configured
DNS Trusted Source not enabled for Trust-any
DNS Trusted Source: Type: IPs : Interface : Idle/Timeout (sec)
…
```

If you have configured the **dns trusted-source** command with non-default settings, it should appear in the running configuration. For example, the following simple policy trusts every DNS server.

```
ciscoasa# show running-config dns
  dns domain-lookup management
  DNS server-group DefaultDNS
   name-server 171.70.168.183 management
  dns trusted-source any
```

Step 4 If the DNS server, lookup, and inspection configurations are correct, and inspection packet count remains 0 because DNS queries are not going through the device, then you cannot obtain traffic classification from DNS snooping. Instead, you must rely on TLS Client Hello and HTTP request host header snooping.

If the DNS cache remains empty, then TLS/HTTP snooping is failing. You can verify the problem by doing packet captures on the ingress interface, to check for the presence of unencrypted SNI fields in TLS Client Hello packets, or the presence of the Host header in HTTP traffic.

If the DNS cache remains empty, you will have to redesign the network to force DNS queries to go through the device. Otherwise, turn off AVC, as it will not be useful on this device.

Troubleshooting Traffic Misclassification

If you find that applications are being inconsistently allowed or blocked, you might have a traffic misclassification problem. This problem can occur in a Content Delivery Network (CDN), where a given IP address might be mapped to different domain names.

To check whether you have this problem, use the **show dns ip-cache** command to check if an IP address is shared by multiple domains.

For example, the following output shows that tiktok.com and fidelity.com are mapped to the same IP address. Because the access control rules are ultimately matched by IP address, if you have separate rules that block one application while allowing the other, the rules will not work as expected. Even if you allow both applications in separate rules, rule and application hit counts will not reflect application usage.

```
ciscoasa(config)# show dns ip-cache
DNS snooping IP cache: 81 in use, 98 most used
Address            Domain                  Idle(sec) Timeout   Hit-count  Source
23.1.106.133       salesforce.com.         2         120       0          DNS
23.67.33.42        fidelity.com.           1         120       42         DNS
                   tiktok.com.             1         120       52         DNS
99.83.221.176      gamespot.com.           8         1495      10         DNS
```

If you see a lot of addresses mapped to multiple applications that you want to handle differently, AVC might not be useful in your particular network. Evaluate whether you should continue using AVC.

Monitoring Allowed and Blocked Applications

You can view the applications that are allowed or blocked by access control rules, and the hit count for the applications. The information includes the access control list that contains the applicable rules.

- **show avc allowed**

```
ciscoasa: show avc allowed
Access-List avc (hitcnt=2)
    Network-Service-Group _business_ (hitcnt=2)
    App Office 365 (hitcnt=1)
    App Microsoft (hitcnt=1)
```

- **show avc blocked**

```
ciscoasa: show avc blocked
Access-List avc (hitcnt=833)
  Network-Service-Group custom_app_blocking (hitcnt=731)
    App TikTok (hitcnt=731)
  Network-Service-Group _gaming_ (hitcnt=102)
    App GameStop (hitcnt=102)
```

Determining the Application Categories for an Application

Applications can appear in more than one category. For a given application, you can check the category using the **show object network-service** command. Knowing the categories can help you design access control allow and block rules, and troubleshoot unexpected allow/block results.

For example, showing the "Fox News" application shows that it belongs to three categories (this can change with the next VDB download, this example is for illustration only).

```
ciscoasa# show object network-service "Fox News"
object network-service "Fox News" dynamic (hitcnt=30)
 description Web Portal for news update.
 app-id 1366
 member of: "_multimedia_(tb/video)_" "_web_services_provider_" "_news_"
…
```

Viewing Application Categories

You can view the list of application categories that are defined as network-service object groups, including their hit counts.

show avc app-category

For example, the following shows a subset of the output:

```
ciscoasa: show avc app-category
gaming: object-group network-service _gaming_ (id=0xfdff0002) (hitcnt=0)
mobile application: object-group network-service _mobile_application_ (id=0xfdff0003)
(hitcnt=0)
social networking: object-group network-service _social_networking_ (id=0xfdff0004) (hitcnt=0)
business: object-group network-service _business_ (id=0xfdff0005) (hitcnt=0)
database: object-group network-service _database_ dynamic (id=0xfdff001d) (hitcnt=0)
education: object-group network-service _education_ dynamic (id=0xfdff001e) (hitcnt=0)
```

Monitoring the DNS Snooping Cache

You can monitor the contents of the DNS IP-to-domain cache created through DNS snooping.

If the cache is empty, it indicates that DNS snooping is not happening on the device. To troubleshoot this problem, see Verify that Application Classification is Happening, on page 14.

Use the **show dns ip-cache** command to monitor the cache. For example:

```
ciscoasa(config)# show dns ip-cache
DNS snooping IP cache: 81 in use, 98 most used
Address            Domain                  Idle(sec) Timeout   Hit-count  Source
23.1.106.133       salesforce.com.         2         120       0          DNS
23.67.33.42        fidelity.com.           1         120       42         DNS
                   tiktok.com.             1         120       52         DNS
99.83.221.176      gamespot.com.           8         1495      10         DNS
```

Each cache entry has a time-to-live (TTL) within the limits of 2 minutes and 24 hours. If DNS resolution returns a TTL less than 2 minutes, the cache entry's TTL is 2 minutes. If the DNS TTL is greater than 24 hours, the cache entry expires after 24 hours. These limits ensure that on the one hand, there is not excessive churn in the cache, and on the other hand, entries do not become stale.

Reloading AVC Network-Service Objects

When a new version of the VDB is downloaded, AVC network-service objects are reconfigured. If you think there is an issue with the AVC network-service objects, you can force the system to reload them. Use the following command to reload the definitions:

network-service reload

For example:

```
ciscoasa# network-service reload
```

Use the **show object network-service detail** command to view all network-service objects, both system defined (AVC) and user configured.

Resetting AVC Hit Counts and Statistics

You can use the following clear commands to reset hit counts and other statistics to 0 without disabling AVC. Resetting statistics can give you a fresh view of what applications are being used in the system.

- **clear avc**

  Clears all AVC related counters, including those for network-service objects and groups. For example:

  ```
  ciscoasa# clear avc
  ```

- **clear object** [**id** *object_name* | **network-service**]

  Clears the counters for network-service objects, both user- and AVC-defined. Use the **id** keyword to specify an object by name. Using the **network-service** keyword provides the same result as using no parameters. For example:

  ```
  ciscoasa# clear object network-service
  ciscoasa# clear object id ns
  ```

- **clear object-group** [**id** *object_name* | **network-service**]

  Clears the counters for object groups, both user-defined and AVC-defined. Use the **id** keyword to specify an object by name. Use the **network-service** keyword to limit the scope to all network-service object groups. For example:

  ```
  ciscoasa# clear object-group network-service
  ciscoasa# clear object-group id nsg
  ```

Monitoring AVC Syslog Messages

The following are the syslog messages associated with the AVC feature:

- 861001: AVC: Creating AVC app directory .app_data failed; Invalid directory.

  The system could not create a directory for the AVC data. Contact Technical Support.

- 861002: AVC: Downloading file from link *URL_link* to directory .app_data succeeded.

  The VDB download succeeded. No action required.

- 861003: AVC: Downloading file from link *URL_link* to directory .app_data failed; no such device.

  The VDB download failed because there was no route to the server. Check your DNS configuration and routing table to ensure names can be resolved and a route exists.

- 861004: AVC: Getting VDB version from file sf.xml failed; Cannot locate the VDB update signature to get VDB version.

  The system downloads a version file to determine if there is a new VDB available for download. The version file is likely corrupted, and the system cannot extract the version number from the file. Contact Technical Support.

- 861005: AVC: Getting VDB file path from file sf.xml failed; Cannot locate the VDB file path signature to get VDB file path.

  The system could not find the VDB file path signature. It is likely that the file is corrupted. Contact Technical Support.

- 861006: AVC: Getting VDB file name from file sf.xml failed; Cannot locate the VDB file name trailer to get VDB file name.

  The system could not extract the VDB file name. It is likely that the file is corrupted. Contact Technical Support.

- 861007: AVC: Loading network service (app) definition file (dynamic-config.json) failed; file not found.

  The system could not create network-service objects for the applications. Try downloading the VDB again. If the problem persists, contact Technical Support.

- 861008: AVC: Loading network service (app) definition file (dynamic-config.json) success.

  The system successfully created network-service objects for the applications. No action required.

- 861009: AVC: Loading app category definition file failed; Opening VDB sqlite.vdb error.

  The system could not open the application category definition file. Try downloading the VDB again. If the problem persists, contact Technical Support.

- 861010: AVC: Loading app category definition file warning; No corresponding NS found for category *name* app id *number*.

  The system could not find any application with the app ID that is specified in the application category. The application likely has been obsoleted. No action required.

- 861011: AVC: Loading app category definition file success.

  The system successfully loaded the application category definition file. No action required.

- 861012: AVC: Installing visibility NSG failed; ERROR: cannot create internal visibility NSG; *error_string*.

  The system could not create the application visibility network-service object group named _avc_visibility_nsg_ or there are errors adding member applications to the visibility network-service group (NSG). The error string provides further explanation about the error. Contact Technical Support.

- 861013: AVC: Installing visibility NSG success.

  The system successfully created network-service object groups for the application categories. No action required.

History for Application Visibility and Control

Feature Name	Platform Releases	Description
Application Visibility and Control for Secure Firewall 6100	9.24(1)	Application Visibility and Control (AVC) makes it possible for you to write access control rules based on applications rather than just IP addresses and ports. AVC downloads the Vulnerability Database (VDB), which creates network-service objects and groups that you can use in access control rules. The objects define various applications, and the groups define application categories, so you can easily block applications or entire classes of connections without specifying IP address and port. We introduced or modified the following commands: **avc**, **avc download vdb**, **clear avc**, **clear object-group**, **network-service reload**, **show avc**, **show service-policy**. In addition, you can no longer enter the **app-id** command as part of a network-service object definition. Supported platforms: Secure Firewall 6100