# ACOS 6.0.7
# DDoS Mitigation Guide (for ADC)

**March, 2025**

Information in this document is subject to change without notice.

## PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: a10-virtual-patent-marking.

## TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

## CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

## DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

## ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

## FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

# Table of Contents

# Getting Started

ACOS provides a suite of security features that allow you to protect your customer traffic:

The following topics are covered:

# Application Access Management

Application Access Management (AAM) is an ACOS security feature that optimizes Authentication, Authorization, and Accounting (AAA) for client-server traffic.

The following topics are covered:



**NOTE:** For more information about AAM, see the *Application Access Management Guide*.

## Login Portal

Provides a sign-on interface. By using a request-reply exchange or using a Web-based form, ACOS obtains the your credentials and uses a backend AAA server to verify these credentials.

## Online Certificate Status Protocol (OCSP)

Provides certificate verification services and eliminates the need to import certificate revocation list (CRL) files to the ACOS device.

The CRLs are maintained on the OCSP responder (server). When a client sends its certificate as part of a request for a secured service, ACOS first sends the certificate to the OCSP responder for verification. After the certificate is verified, the client can access secured services.

## Authentication Relay

Offloads your AAA servers. ACOS contacts the backend AAA servers on behalf of the clients, and after a server responds, ACOS caches the reply and uses this reply for

subsequent client requests.

## AAA Health Monitoring and Load Balancing

Load balances authentication traffic among a group of AAA servers. ACOS supports custom health checks for LDAP, RADIUS, Kerberos, and OCSP.

# Online Certificate Status Protocol

Online Certificate Status Protocol (OCSP) is a network component that provides certificate verification services.

OCSP is an efficient alternative to CRLs, which is also supported by ACOS. To use CRLs with ACOS, you must import the CRL files into the ACOS device. If you use OCSP, ACOS can also send certificate verification queries to external OCSP servers (generally called *responders*). This process only occurs when a client sends a certificate as part of a request to set up a secure session to a server application that is managed by ACOS.

**NOTE:** For more information about OSCP, see: **Checking Client Certificates Using OCSP** in the *SSL Configuration Guide* and **AAM with OCSP** in the *Application Access Management Guide*.

# DDoS Mitigation

Distributed Denial of Service (DDoS) is a type of DoS attack where multiple systems that are infected with a Trojan or malware are, in turn, used to target a particular system. This process causes a denial of service. If a hacker (attacker) mounts an attack from one host, this is classified as a DoS attack. In a DDoS attack, many systems are used simultaneously to launch attacks against a remote system.

ACOS includes filters that check traffic for IP anomalies that can indicate a DDoS attack.

**NOTE:** For more information about DDos Mitigation, see IP Anomaly Filtering.

# Attack Detection and Prevention using ZBAR

The ADC has combined multiple infrastructural capabilities to create an enhanced solution that identifies volumetric and IOT DDoS attacks on the SLB virtual port. It employs mitigation policies to provide excellent application responsiveness for the good actors. The bad sources are dropped or rate-limited based on their computed threat score. The **attack-detection** command needs to be set to enable this feature.

This solution employs the following infrastructure capabilities:

- Analytics infrastructure – It adaptively baselines multiple traffic metrics and creates a monitoring entity that observes and determines the traffic baselines and sets attack thresholds. On attack detection, it triggers the ZBAR infrastructure.
- ZBAR (Zero Day source behavior Attack Recognition) infrastructure - It dynamically classifies the incoming traffic based on real-time traffic conditions. The ZBAR framework performs clustering based on multiple metrics from the sources hitting the attacked Destination/Destination-Service. By mapping the sources to these metrics, ZBAR determines the miscreant source/sources, along with the associated confidence score, suspected of causing a DDoS attack.
- Packet capture infrastructure – It automatically captures a few packets from the deemed attackers and stores them as evidence and signature extraction. You can view these captured packets using the following show command,

  ```
  show visibility packet-capture packet-capture-files
  ```

  The captured packets for the bad sources can also be exported using the following command,

  ```
  export visibility pktcapture-file file
  ```

To enable this feature, refer to the following configuration:

```
ACOS (config)# slb virtual-server vip1 12.12.12.203
ACOS (config-slb vserver)# port 80 tcp
ACOS (config-slb vserver-vport)# attack-detection
```

Additionally, the following show commands are added to view ZBAR information:

- `show visibility zbar dest`
- `show visibility zbar dest bad-sources`

For more information, see the *Application Delivery Controller Guide*.

## Single CPU Attack Prevention

The CPU Round Robin feature is used to mitigate the effects of Denial of Service (DoS) attacks that target a single CPU on the ACOS device. The command `system cpu-load-sharing` is used to configure thresholds for CPU load sharing. If a threshold exceeds, CPU load sharing is activated, and additional CPUs are enlisted to help process the traffic and relieve the burden on the targeted CPU. A round robin algorithm distributes packets across all the other data CPUs on the device. Load sharing will remain in effect until traffic no longer exceeds the thresholds that originally activated the feature.

**NOTE:** For more information about the command `system cpu-load-sharing`, see *Command Line Reference* guide.

## Policy-Based SLB

Policy-based SLB (PBSLB) allows you to “black list” or “white list” individual clients or client subnets. Based on actions that you specify, ACOS will allow (white list) or drop (black list) traffic from specific client hosts or subnets in the list.

**NOTE:** For more information about policy-based SLB, see Policy-based SLB.

## SYN Cookies

SYN cookies provide protection against a common type of DDoS attack, the TCP SYN flood attack. The attacker sends a high volume of TCP-SYN requests to the target device, but the attacker does not reply to SYN-ACKs to complete the three-way handshake for any of the sessions. The purpose of the attack is to consume the target’s resources with half-open TCP sessions.

When SYN cookies are enabled, the ACOS device can continue to serve legitimate clients during TCP SYN flood attacks, while preventing illegitimate traffic from consuming system resources.

**NOTE:** For more information about SYN cookies, see SYN Cookies.

## IP Limiting

IP limiting provides a enhanced implementation of the source IP connection limiting and connection-rate limiting feature that was available in earlier releases.

**NOTE:** For more information about IP limiting, see IP Limiting.

## ICMP Rate Limiting

ICMP rate limiting protects against ICMP-based or ICMPv6-based DoS attacks, such as Smurf attacks, which consist of floods of spoofed broadcast ping messages. ICMP rate limiting monitors the rate of ICMP traffic and drops ICMP packets when the configured thresholds have been exceeded.

**NOTE:** For more information about ICMP rate limiting, see ICMP Rate Limiting.

## Web Application Firewall

ACOS provides additional security for your Web servers with the Web Application Firewall (WAF) feature. WAF filters communication between end-users and Web applications to protect Web servers and sites from unauthorized access and malicious programs.

This new layer of security examines the following types of traffic to safeguard against Web attacks and protect sensitive information hosted on Web servers:

- Incoming user requests
- Output from Web servers
- Access to Web site content

**NOTE:** Fore more information about WAF, see the *Web Application Firewall Guide*.

## Slowloris Prevention

In addition to the WAF, ACOS includes an HTTP security option that prevents Slowloris attacks, in which the attacker attempts to consume resources on the target system with incomplete HTTP request headers.

**NOTE:** For more information about Slowloris prevention, see HTTP Slowloris Prevention.

## DNS Application Firewall

DNS Application Firewall (WAF) filters for malformed queries. The DAF also protects against "**any**" queries for all DNS records. An "any" query is a request for a DNS server to send copies of all of its DNS records. Because this type of query can heavily consume DNS resources, it is sometimes used as a DDoS attack.

**NOTE:** For more information about DAF, see DNS Application Firewall.

## DNSSEC

ACOS supports DNS Security Extensions (DNSSEC). In Global Server Load Balancing (GSLB) deployments, you can use DNSSEC with Hardware Module Security (HSM) to dynamically secure DNS resource records for GSLB zones.

**NOTE:** For more information about DNSSEC, see DNSSEC Support.

**NOTE:** The ACOS also supports DNS caching for DNSSEC, but DNSSEC support for caching does not require GSLB.

## SSL Insight

SSL Insight (SSLi) provides high-performance SSL decryption and re-encryption. When used in conjunction with third-party traffic inspection devices, SSLi adds content-

level security.

SSLi decrypts SSL-encrypted client traffic and sends the decrypted traffic to a third-party traffic inspection device. Traffic that is permitted by the traffic inspection device is re-encrypted by ACOS and forwarded to its destination.

**NOTE:** For more information about SSL Insight, see "**SSL Insight**" in the *SSL Configuration Guide*.

## Geo-location-based VIP Access

Geo-location-based VIP access controls the access to a VIP based on the client's location. You can configure ACOS to perform one of the following actions for traffic from a client, depending on the location of the client:

- Drop the traffic
- Reset the connection
- If configured by using a black/white list, send the traffic to a specific service group

ACOS determines a client's location by looking up the client's subnet in the geo-location database that is used by Global Server Load Balancing (GSLB).

**NOTE:** For more information about Geo-location-based VIP access, see Location-Based VIP Access.

# IP Anomaly Filtering

ACOS helps you detect and mitigate Distributed Denial of Service (DDoS) attacks. One of the features, IP anomaly filtering, can protect against numerous types of attacks.

The following topics are covered:

# Overview of IP Anomaly Filtering

IP anomaly filtering detects and drops packets that contain the common signatures of DDoS attacks.

The following topics are covered:



## IP Anomaly Filters

Users can enable the following IP anomaly filters. This section has the following sub-sections:

The following topics are covered:



### Frag

Drops all IP fragments, which can be used to attack hosts that run IP stacks with known vulnerabilities in their fragment reassembly code.

### IP-option

Drops all packets with IP options.

### Land-attack

Drops spoofed SYN packets that contain the same IP address as the source and destination. These packets can be used to launch an "`IP land attack`".

### Ping-of-death

Drops all jumbo ICMP packets, which are also known as "`ping of death`" packets.

### TCP-no-flag

Drops all TCP packets that have no TCP flags set.

### TCP-SYN-FIN

Drops all TCP packets in which both the SYN and FIN flags are set.

### TCP-SYN-frag

Drops incomplete (fragmented) TCP Syn packets, which can be used to launch TCP Syn flood attacks.

## Threshold

The threshold specifies the number of times the anomaly is allowed to occur in a client's connection requests.

If system-wide PBSLB is configured, ACOS applies the policy's over-limit action to clients that exceed the threshold. The range for the threshold value is 1-127 occurrences of the anomaly, and the default value is 10.

**NOTE:** The thresholds are not tracked by PBSLB policies that are bound to individual virtual ports.

## SOCKSTRESS_CHECK Session State

When the ACOS device checks a data packet against the new IP anomaly filters, the client's session is in the SOCKSTRESS_CHECK state. You might see this state if you are viewing debug output for the client's session.

## Implementation Notes

Consider the following implementation notes when you want to apply IP anomaly filtering:

- DDoS mitigation feature is supported on the FTA-based Thunder series hardware models, such as TH4440, TH7655S, etc.
- DDoS protection is software-based on other models.
- DDoS detection applies only to Layer 3, Layer 4, and Layer 7 traffic. However, Layer 4 and Layer 7 DDoS applies only to software releases that support Server Load Balancing (SLB).
- All IP anomaly filters, except “IP-option”, apply to IPv4 and IPv6. The “IP-option” filter applies only to IPv4.
- For Thunder 3030S, Thunder 1030S, and Thunder 930 models, all IP packets longer than 32000 bytes are dropped. For other models, IP packets that are longer than 65535 bytes are dropped.

# Configuring IP Anomaly Filtering

By default, all the IP anomaly filters that are described in this chapter are disabled. You can enable individual IP anomaly filters, on a system-wide basis.

The following topics are covered:



## Using the GUI to Configure IP Anomaly Filtering

To use the GUI, navigate to **Security >> DDoS Protection** and select the anomaly for which you want to enable protection.

## Using the CLI to Configure IP Anomaly Filtering

To enable IP anomaly filters from the CLI, use the `ip anomaly-drop` command.

For example, the following command enables DDoS protection against ping-of-death attacks:

```
ACOS(config)# ip anomaly-drop ping-of-death
```

Refer to the "ip anomaly-drop" command in the *Network Configuration Guide* for more information about this command.

# Displaying IP Anomaly Statistics

This section describes how to view IP anomaly statistics.

The following topics are covered:



## Using the GUI to Display IP Anomaly Statistics

Navigate to **ADC >> Statistics >> Switch**.

**NOTE:** For more information, see the *online Help*.

## Using the CLI to Display IP Anomaly Statistics

To view system-wide Layer 4 SLB traffic statistics, use the **`show slb l4`** command. This command provides an overview of L4 traffic, active connections, and general dropped packets.

To display detailed information on packets dropped due to IP anomalies, use the **`show ip anomaly-drop statistics`** command. This command specifically shows packet drops caused by malformed packets, protocol violations, SYN floods, and other IP anomalies.

**NOTE:** SNMP monitoring is also supported for IP anomaly drop statistics. To enable this capability, use **`snmp-server enable schema-agent`** command to enable the new CM subagent.

To clear all Layer 4 SLB statistics, including the IP anomaly counters, enter the **`clear slb l4`** command.

**NOTE:** For more information about these commands, see *Command Line Interface Reference Guide*.

# Policy-based SLB

This chapter helps you understand and configure policy-based SLB (PBSLB).

The following topics are covered:

# Overview

ACOS allows you to “black list” or “white list” individual clients or client subnets. White list traffic is allowed, and black list traffic is dropped from specific client hosts or subnets in the list.

For white list traffic, you can specify the service group to use. You also can specify the action that will be taken (drop or reset) on new connections that exceed the configured connection threshold for the client address.

## Example

The user can configure ACOS to respond to DDoS attacks from a client by dropping excessive connection attempts from the client.

You can apply PBSLB on a system-wide basis. If Server Load Balancing (SLB) is supported, you also can apply PBSLB on individual virtual ports.

**NOTE:** ACOS also allows policy templates to be applied at the virtual-server level. However, PBSLB does not take effect if you apply the policy template at the virtual-server level. Only class lists are supported at the virtual-server level. To use PBSLB, you must apply the policy template globally or on individual virtual ports.

**NOTE:** If a connection limit is specified in a black/white list, the ACOS device does not support using the list for system-wide PBSLB and for PBSLB on an individual virtual port. In this case, the ACOS device may increase the current connection counter more than once, which results in a much lower connection limit than the configured value. To resolve this issue, you should use separate black/white lists.

# Configuring a Black/White List

The following sections are described in this topic:

The following topics are covered:

# Configuration Details and Examples

Client IP lists, such as black/white lists, can be configured on an external device and imported to the ACOS device or can be entered in the GUI. The actions to take on the addresses in the list are specified on the ACOS device. A black/white list can contain up to 8 million individual host addresses and up to 64,000 subnet addresses.

For each IP address (host or subnet) in a black/white list, you can add a row by using the following syntax:

```
ipaddr [/network-mask] [group-id] [#conn-limit] [;comment-string]
```

The syntax is defined in the following way:

Table 1 : Black/White List

| Parameter | Description |
|---|---|
| *ipaddr* | Host or subnet address of the client. |
| *network-mask* | Optional network mask length. The default is 32, which means that the address is a host address. |
| *group-id* | Number between 1 and 31 in a Black/White list that identifies a group of IP host or subnet addresses in the list. In a PBSLB policy template on the ACOS device, you can map the group to one of the following actions:<br>Drop the traffic<br>Reset the connection<br>Send the traffic to a specific service group<br>The default group ID is 0, which means that no group is assigned. |

Table 1 : Black/White List

| Parameter | Description |
|---|---|
| *#conn-limit* | Maximum number of concurrent connections that are allowed from the client. By default, there is no connection limit. If you decide to set a limit, the valid range is between 1 and 32767. On the ACOS device, you can specify whether to reset or drop new connections that exceed this limit.<br><br>The # is required only if you do not specify a *group-id*. |
| *comment-string* | Comment; everything to the right of the semi-colon (;) is ignored by the ACOS device when it parses the file. |

**NOTE:** The *conn-limit* is a coarse limit. The larger the number you specify, the more coarse the limit.

### Example

If you specify 100, the ACOS device limits the total connections to 100.

- As another example, if you specify 1000, the device limits the connections to a maximum of 992 connections.
- If the number in the file is larger than the supported maximum limit value, the parser uses the longest set of digits in the number that you enter that makes a valid value.

### Example

- If the file contains 32768, the parser uses 3276 as the value.
- As another example, if the file contains 111111, the parser will use 11111 as the value.

## Example Black/White List

The following text is a sample black/white list:

```
10.10.1.3 4; blocking a single host. 4 is the drop group
10.10.2.0/24 4; blocking the entire 10.10.2.x subnet
```

```
192.168.1.1/32 #20 ; 20 concurrent connections max, any group ok
192.168.4.69 2 #20 ; assign to group 2, and allow 20 max
```

The first row assigns a specific host to group 4. On the ACOS device, the drop action is assigned to this group, which black lists the client.

The second row black lists an entire subnet by assigning it to the same group (4).

The third row sets the maximum number of concurrent connections for a specific host to 20.

The fourth row assigns a specific host to group 2 and specifies a maximum of 20 concurrent connections.

**NOTE:** The ACOS device allows up to three parser errors when reading the file but stops reading after the third parser error.

## Dynamic Black/White-list Client Entries

The ACOS device supports dynamic client entries. You can configure this feature by adding the client address 0.0.0.0/0 (wildcard address) to the black/white list that is used by the system-wide PBSLB policy.

When a client sends an HTTP or HTTPS connection request, the ACOS device checks the system-wide PBSLB policy's black/white list for the client's IP address, with one of the following results:

- If there is no entry for the client, he ACOS device creates a dynamic entry for the client's host address.
- If there is a dynamic entry for the client, the ACOS device resets the timeout value for the entry. (Dynamic entry aging is described below.)

**NOTE:** If there is a static entry for the client's host or subnet address, the static entry is used instead.

The following is an example of a wildcard address in a black/white list:

```
0.0.0.0/0 1 #20
```

In this example, the clients who do not match a static entry in the list are assigned to group 1 and are limited to 20 concurrent connections.

The ACOS device supports up to 8 million dynamic client entries for system-wide PBSLB. Once this limit is reached, the ACOS device no longer track connections or anomaly counters for additional clients.

## Connection Limit for Dynamic Entries

For dynamic entries in a system-wide PBSLB policy's black/white list, the connection limit in the list applies to each client.

In the example above, each client that has a dynamic entry in the black/white list will be allowed to have a maximum of 20 concurrent connections.

## Aging of Dynamic Entries

When the ACOS device creates a dynamic black/white list entry for a client, the device also sets the timeout for the entry. The timeout value for the dynamic entry decreases until the timeout reaches 0 or the client sends a new HTTP or HTTPS connection request.

If the client sends a new HTTP or HTTPS connection request, the timeout is reset to its full value. If the timeout reaches 0 and the client does not have active connections, the dynamic entry is removed. However, if the client has an active connection, the dynamic entry is not removed until the client's connection ends. You can set the timeout to 1-127 minutes, and the default is 5 minutes.

If client-lockup is enabled, the timeout for a locked up client does not begin decreasing until the lockup expires.

## Wildcard Address Support in PBSLB Policies Bound to Virtual Ports

Dynamic client entries are supported only for system-wide PBSLB policies.

You can add a wildcard address (0.0.0.0/0) to a black/white list that is used by a virtual port's PBSLB policy. The group ID and connection limit that are specified for the wildcard address are applied to clients that do not match a static entry in the list.

Consider the following limitations:

- The ACOS device does not create dynamic entries in the list.
- The connection limit applies collectively to all clients that do not have a static entry in the list.

# Configuring System-wide PBSLB

The following topics are covered:



## Options for System-wide PBSLB Policies

System-wide PBSLB policies provide the following options:

- Dynamic black/white-list client entries
- Client lockup
- IP anomaly checking and tracking, using IP anomaly filters

These options are not available in policies that are applied to individual ports.

## Using the GUI to Configure System-wide PBSLB

To configure a system-wide PBSLB policy using the GUI, do the following:

1. Configure the PBSLB settings in an SLB policy template.
   a. Navigate to **ADC >> Template >> L7**.
   b. Click **Create** and select **Policy** from the drop-down list.
   c. Specify a policy name; for example, **pol1**.
   d. Expand the BW List section, and configure the Black/White list settings as desired.
   e. Click **OK**.
2. Apply the policy template at the system level.
   a. Navigate to **ADC >> SLB >> Global**.
   b. In the System template policy field, select **pol1** from the drop-down list.
   c. Click **Update**.

## Using the CLI to Configure System-wide PBSLB

To configure a system-wide PBSLB policy using the CLI, do the following:

1. Configure the PBSLB settings in an SLB policy template.

   The following example drops any connections from clients exceeding one of the following limits:

   - The connection limit that is configured in the specified in the Black/White list.
   - The threshold of any of the new IP anomaly filters.

     Logging is enabled and messages are generated two minutes.

```
ACOS(config)# slb template policy pol1
ACOS(config-policy)# bw-list id 1 drop logging 2
ACOS(config-policy)# bw-list over-limit lockup 5 logging 2
ACOS(config-policy)# exit
ACOS(config)#
```

2. Apply the policy template at the system level:

```
ACOS(config)# system template policy pol1
```

## Displaying and Clearing System-wide PBSLB Information

To display information for system-wide PBSLB, enter the **show pbslb system** or **show pbslb client** commands.

To clear PBSLB information, use the **clear pbslb system** or **clear pbslb client** commands.

Use the **entry** option with the **clear pbslb client** command to clear both statistical counters and client entries; without this option, only the statistical counters are cleared.

# Configuring PBSLB for Individual Virtual Ports

The following topics are covered:



## Configuration Details

You can configure PBSLB parameters for virtual ports by configuring the settings on individual ports or by configuring a PBSLB policy template and binding the template to individual virtual ports.

**NOTE:** This feature is supported only in software releases that support Server Load Balancing (SLB).

These steps assume that the real servers, service groups, and virtual servers have already been configured.

To configure PBSLB:

1. Configure a black/white list remotely or on the ACOS device.
2. If you configure the list remotely, import the list to the ACOS device.
3. Optionally, modify the sync interval for the list.
4. ACOS regularly synchronizes with the list to ensure that the ACOS version is current.
5. Configure PBSLB settings.

You can configure a policy template and bind the template to virtual ports or configure the following settings on individual virtual ports:

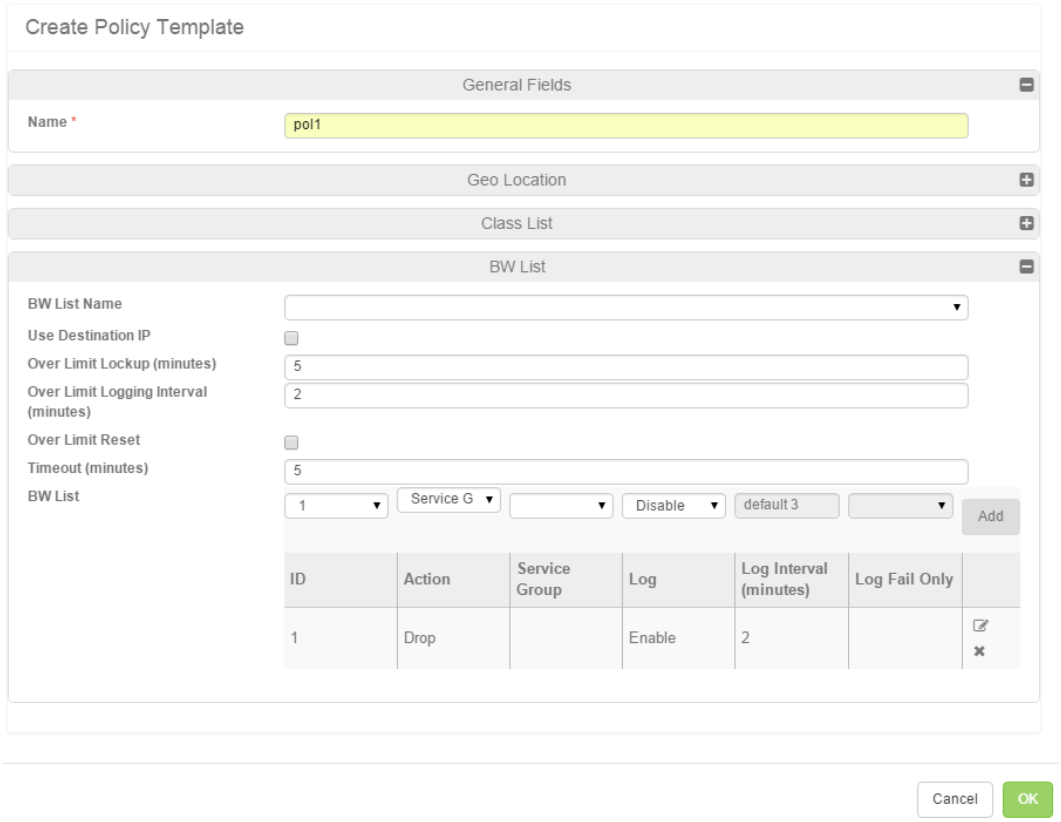
- Specify the black/white list.
- Optionally, map each group ID that used in the list to one of the following actions:
  - Send the traffic to a specific service group.
  - Reset the traffic.
  - Drop the traffic.
- Optionally, change the action (drop or reset) that ACOS will take on connections that exceed the specified limit.
- Optionally, if necessary, change the client address matching from source IP matching to destination IP matching.

## Using the GUI to Configure PBSLB for Individual Virtual Ports

To configure a PBSLB policy for individual virtual ports using the GUI, do the following:

1. Configure the PBSLB settings in an SLB policy template.
   a. Navigate to **ADC >> Template >> L7**.
   b. Click **Create** and select **Policy** from the drop-down list.
   c. Specify a policy name; for example, **pol1**.
   d. Expand the BW List section, and configure the Black/White list settings as desired.

e. Click **OK**.




2. Apply the policy template at the virtual port level.
   a. Navigate to **ADC >> SLB >> Virtual Servers**.
   b. Click **Edit** in the Actions column for an existing virtual server.
   c. On the Update Virtual Server page, click **Edit** in the Actions column for an existing virtual port.
   d. On the Update Virtual Port page, expand the Templates section.
   e. Select the desired policy template from the drop-down list in the Template Policy field.
   f. Click **Update**.

# Using the CLI to Configure PBSLB for Individual Virtual Ports

The following commands import black/white list “sample-bwlist.txt” to the ACOS device:

```
ACOS(config)# import bw-list sample-bwlist tftp://myhost/TFTP-Root/ACOS_
bwlists/sample-bwlist.txt
ACOS(config)# show bw-list

Name              Url                                  Size(Byte)    Date
-------------------------------------------------------------------------
----
sample-bwlist     tftp://myhost/TFTP-Root/ACOS_          N/A           N/A
                  bwlists/sample-bwlist.txt
Total: 1
```

The following commands configure a PBSLB template and bind it to a virtual port:

```
ACOS(config)# slb template policy bw1
ACOS(config-policy)# bw-list name bw1
ACOS(config-policy)# bw-list id 2 service-group srvcgroup2
ACOS(config-policy)# bw-list id 4 drop
ACOS(config-policy)# exit
ACOS(config)# slb virtual-server PBSLB_VS1 10.10.10.69
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy bw1
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

The following commands displays PBSLB information:

```
ACOS(config-slb vserver-vport)# show pbslb
Total number of PBSLB configured: 1
Virtual Server Port Blacklist/whitelist GID Connection # (Establish Reset
Drop)
-------------------------------------------------------------------------
----
```

```
PBSLB_VS1       80    sample-bwlist        2    0          0          0
                                           4    0          0          0
PBSLB_VS2       80    sample-bwlist        2    0          0          0
                                           4    0          0          0
```

# Configuration Example for Sockstress Attack Protection

You can use system-wide PBSLB with IP anomaly filters to protect against Sockstress attacks, which is a type of DDoS attack.

In this example, the ACOS device drops all new connection attempts from a client if one of the following conditions occur:

- The client already has 20 active connections and attempts to open a new HTTP or HTTPS connection.
- The client exceeds any of the IP anomaly thresholds.

The lockup period is set to 5 minutes, to continue enforcing the over-limit action for 5 minutes after the over-limit action is triggered. The timeout for dynamic black/white list entries is set to 2 minutes.

This example uses the following black/white list:

```
0.0.0.0/0 1 #20
```

# PBSLB Statistics Display

The following command displays system-wide statistics for the new IP anomaly filters:

```
ACOS(config)# show slb l4
Total
-----------------------------------------------------------------
IP out noroute              20061
TCP out RST                 0
```

```
TCP out RST no SYN       0
...
Anomaly out of sequence 225408
Anomaly zero window     225361
Anomaly bad content     224639
```

The following command displays statistics for the system-wide PBSLB policy:

```
ACOS(config)# show pbslb system
System    B/W list: bwlist-wc
Virtual Server Port Blacklist/whitelist GID Connection # (Establish Reset
Drop)
------------------------------------------------------------------------
------
System              bwlist-wc           1   12         0          0
                                        2   0          0          0
```

The following command displays summary statistics for individual black/white-list clients:

```
ACOS# show pbslb client
                                    GID = Group ID, S/D = Static or dynamic
entry
              Out-s = Out of sequence, Zero-w = Zero window, Bad-c = Bad
content
IP                 S/D GID Conn-limit Curr-conn Age   Lockup Out-s Zero-w
Bad-c
-----------------+---+---+----------+---------+-----+------+-----+------
+----
40.40.40.168   /32  D  1   20         5         120   0      0     5
5
40.40.40.169   /32  D  1   20         6         0     5      0     6
6
40.40.40.170   /32  D  1   20         6         0     5      0     6
6
40.40.40.171   /32  D  1   20         6         0     5      0     6
6
```

```
40.40.40.172    /32   D  1    20          6          0      5      0      6
6
40.40.40.173    /32   D  1    20          2          120    0      0      2
2
40.40.40.174    /32   D  1    20          5          120    0      0      5
5
40.40.40.175    /32   D  1    20          5          120    0      0      5
5
40.40.40.160    /32   D  1    20          5          120    0      0      5
5
40.40.40.161    /32   D  1    20          6          120    0      0      6
6
40.40.40.162    /32   D  1    20          6          0      5      0      6
6
40.40.40.163    /32   D  1    20          6          0      5      0      6
6
40.40.40.164    /32   D  1    20          6          0      5      0      6
6
40.40.40.165    /32   D  1    20          5          120    0      0      5
5
```

The Age column indicates how many seconds are left before a dynamic entry ages out. For clients who are currently locked out of the system, the value in the Lockup column indicates how many minutes the lockup will continue. For locked up clients, the age value is 0 until the lockup expires. After the lockup expires, the age is set to its full value. In this example, the lockup value is 120 seconds.

The following command displays detailed statistics for a specific black/white-list client:

```
ACOS# show pbslb client 40.40.40.168
IP address:                                    40.40.40.168
Netmask length:                                32
Type:                                          Dynamic
Group ID:                                      1
Connection limit (0 = no limit):               1984
Current connection:                            6
Age:                                           0 second
Lockup time:                                   5 minute
```

```
Out of sequence:                                0
Zero window:                                    6
Bad content:                                    6
```

# SYN Cookies

This chapter describes the SYN-cookie feature and how it helps protect ACOS devices against disruptive SYN-based flood attacks.

The following topics are covered:

# Overview of SYN Cookies

SYN cookies protect against TCP SYN flood attacks. When SYN cookies are enabled, the ACOS device can continue to serve legitimate clients during these attacks, while preventing illegitimate traffic from consuming system resources.

The following topics are covered:



## SYN Flood Attacks

During a TCP SYN flood attack, an attacker sends many TCP SYN Requests to a network device, such as a server. The server replies with a standard SYN-ACK message. However, rather than reply to this attempt at establishing a 3-way handshake with the standard ACK, an attacker ignores the reply and creates a “half-open” TCP connection. System resources are consumed because the device waits for a response from the client that never arrives.

Under large-scale attacks, excessive half-open connections cause a network device’s TCP connection queue to become full. This over-subscription prevents the device from establishing new connections with legitimate clients.

## Identifying SYN Flood Attacks

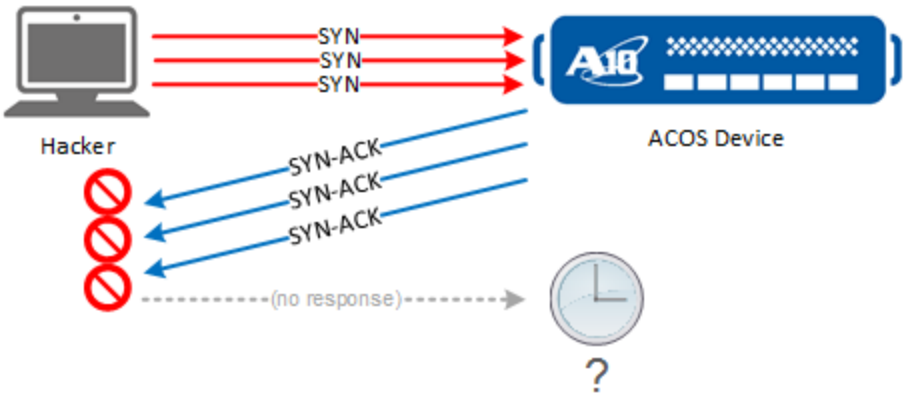
The graphics in this section illustrate how the ACOS device determines whether a particular TCP connection is from a legitimate request or if it is part of a SYN flood attack.

The SYN-ACK Handshake (Legitimate Client) depicts a typical 3-way TCP handshake, which includes a SYN request from the client, the SYN-ACK reply from the ACOS device, and finally, an ACK from the client to the ACOS device.

Figure 1 : SYN-ACK Handshake (Legitimate Client)


However, SYN flood attacks (SYN-ACK Handshake (Hacker) ) can cripple a network by sending multiple SYN requests to a network device. The device responds to these SYN requests with SYN-ACKs and waits for responses from the client that never arrive. These bogus requests create many “half-open” sessions, which wastes system memory and other system resources. The state of being oversubscribed reduces the device’s free resources, which prevents it from accepting requests from legitimate clients.

Figure 2 : SYN-ACK Handshake (Hacker)




Enabling SYN cookies mitigates the damage caused by such DoS attacks by preventing the attacks from consuming system resources.

TCP connections for which the ACOS device did not receive an ACK from the client is identified as belonging to a SYN flood attack, and this information is displayed with the counter in the output of the show command.

## ACOS SYN-cookie Protection

By enabling SYN cookies, the ACOS device's TCP connection queue is prevented from filling up during TCP SYN flood attacks. When a client sends an SYN request, the ACOS device responds with a SYN cookie. This response is a special type of SYN ACK message.

SYN cookies prevent hackers from consuming excessive system resources by encoding the necessary state information for the client connection in a TCP sequence number. Rather than storing state information for each TCP session, the sequence number in the SYN cookie acts as a shorthand, which allows the ACOS device to compress much of the session information into a smaller amount of data.

This sequence number is sent to the client as a SYN-ACK packet. When a legitimate client receives this information, it replies with an ACK that contains the sequence number plus 1.

When the SYN ACK that contains the sequence number from the client is received, the ACOS device reconstructs the connection information and establishes a connection with that client.

If the SYN Request is part of an attack, the attacker does not send an ACK to the ACOS device. The ACOS device sends a SYN cookie, but the attacker does not receive it (or may choose to ignore it), and the ACOS device does not establish a connection.

## Dynamic SYN Cookies

You can configure on and off thresholds for SYN cookies. When there are no TCP SYN attacks, the TCP options are preserved.

You can configure the following dynamic SYN cookie options:

- On-threshold – specifies the maximum number of concurrent half-open TCP connections that are allowed on the ACOS device, before SYN cookies are enabled. If the number of half-open TCP connections exceeds the on-threshold value, the ACOS device enables SYN cookies. You can specify 0-2147483647 half-open connections.
- Off-threshold – specifies the minimum number of concurrent half-open TCP connections for which to keep SYN cookies enabled. If the number of half-open TCP

connections falls below this level, SYN cookies are disabled. You can specify 0-2147483647 half-open connections.

By default, hardware-based SYN cookies are disabled. When the feature is enabled, there are no default settings for the on- and off-threshold. If you omit the on-threshold and off-threshold options, SYN cookies are enabled and are always on, regardless of the number of half-open TCP connections on the ACOS device.

**NOTE:** It may take up to 10 milliseconds for the ACOS device to detect and respond to crossover of either threshold.

## SYN Cookie Buffering

SYN Cookie Buffering optimizes performance by increasing the amount of buffers that are allocated to TCP connections when system memory usage is low and reducing the number of buffers when system memory usage is high.

When SYN cookies are enabled, the ACOS device allocates 10 buffers to each TCP connection, and by default, offers a TCP window size of 8000.

When memory usage increases and system resources are scarce, the number of buffers that are reserved for each TCP connection gradually reduces from 10 buffers to 1 buffer per TCP connection. The window size also reduces during this process.

SYN Cookie Buffering is automatically enabled when SYN cookies are enabled. By default, 10 buffers are allocated to each TCP connection. Instead being dropped and requiring later re-transmission, the packets are stored in the ACOS device's memory and forwarded to the real server when the back-end connection is available.

**NOTE:** This feature is not supported with SLB fast-path processing.

## SACK and MSS with Software-based SYN-cookies

Software-based SYN cookies is an optional feature that is available on certain AX models at the configuration level for virtual ports. The ACOS device bases Selective Acknowledgment (SACK) support, and the maximum segment size (MSS) setting, in software-based SYN cookies on server replies to TCP health checks that are sent to the servers.

The following topics are covered:

### SACK

The ACOS device includes the Sack-Permitted option in TCP SYN health check packets sent to servers.

- If all of the up servers in the service group reply with a TCP SYN-ACK that contains a SACK option, the ACOS device uses SACK with the software-based SYN-cookie feature for all servers in the service group.
- If any of the up servers in the service group do not send a SACK option, the ACOS device does not use SACK with the software-based SYN-cookie feature for any servers in the service group.

### MSS

The lowest MSS value that is supported by a server in the service group is the MSS value that is used by the ACOS device for software-based SYN-cookies.

## Configuring SYN Cookies

The following sections describe how to enable SYN-cookie support and configure advanced features.

The following topics are covered:



## Enabling SYN-cookie Support

The following topics are covered:

## Details

Depending on the Thunder or AX model, you can use hardware-based SYN cookies or software-based SYN cookies:

- Hardware-based SYN cookies can be globally enabled and applied to all virtual server ports that are configured on the device.
- Hardware-based SYN cookies are available on FTA devices. See the FTA Devices section on the A10 Hardware Install Guides website for a list of FTA Thunder and AX devices.
- Software-based SYN cookies can be enabled on individual virtual ports. This version of the feature is available on all AX models.

Consider the following information:

- Hardware-based SYN cookies are a faster, easier-to-configure alternative to the software-based SYN cookie feature available on all AX platforms.

If your AX model supports hardware-based SYN cookies, A10 Networks recommends that you use the hardware-based version of the feature instead of the software-based version.

If both hardware-based and software-based SYN cookies are enabled, only hardware-based SYN cookies are used. Although software-based SYN cookies can be enabled, they are not used.

If Application Delivery Partitioning (ADP) is configured, hardware-based SYN cookies apply to all partitions. The feature is not partition-aware.

- If the target VIP is in a different subnet from the client-side router, use of hardware-based SYN cookies requires some additional configuration.

**NOTE:** For more information, see Configuration with Target VIP and Client-side Router in Different Subnets.

- Software-based SYN cookies are supported only in software releases that support SLB.

### FTA Models

To enable hardware-based SYN cookies on ACOS models that feature FTAs, use the **syn-cookie enable** command at the global configuration level:.

The command in the following example enables dynamic-based SYN cookies when the number of concurrent half-open TCP connections exceeds 50000 and disables SYN cookies when the number falls below 30000:

```
ACOS(config)# syn-cookie enable on-threshold 50000 off-threshold 30000
```

### Non-FTA Models

To enable software-based SYN cookies, use the **syn-cookie** command at the virtual-port level. For example:

```
ACOS(config)# slb virtual-server vip1
ACOS(config-slb vserver)# port 80 tcp
ACOS(config-slb vserver-vport)# syn-cookie
```

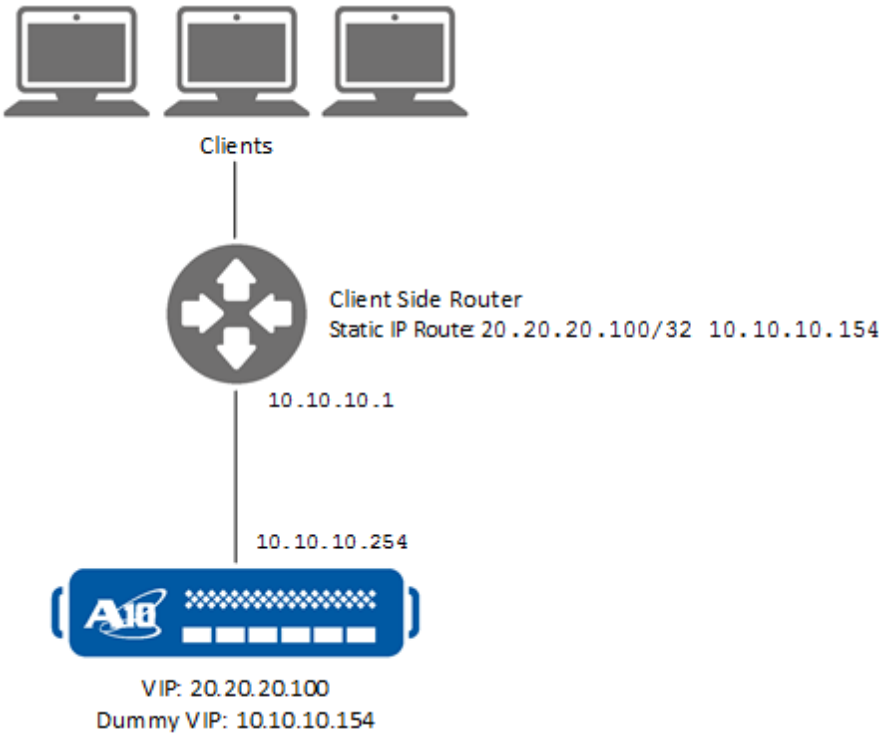
## Configuration with Target VIP and Client-side Router in Different Subnets

Usually, the target VIP in an SLB configuration is in the same subnet as the client-side router. However, if the target VIP is in a different subnet, to use hardware-based SYN cookies, configure the following items:

- On the ACOS device, configure a “dummy” VIP that is in the same subnet as the client-side router.
- On the client-side router, configure a static route to the VIP by using the dummy VIP as the next hop.

Hardware-based SYN Cookies – Target VIP and Client-Side Router in Different Subnets is an example of this deployment.

Figure 3 : Hardware-based SYN Cookies – Target VIP and Client-Side Router in Different Subnets




The following commands configure hardware-based SYN cookies on the ACOS device:

```
ACOS(config)# slb virtual-server dummyvip 10.10.10.154
ACOS(config-slb vserver)# exit
ACOS(config)# syn-cookie
```

**NOTE:** If VRRP-A is configured, add both the target VIP and the dummy VIP to the same VRID so these VIPs will fail over as a unit.

## Modifying the Threshold for TCP Handshake Completion

To modify the threshold for TCP handshake completion, use the **`ip tcp syn-cookie threshold`** global configuration command.

For example, to set the threshold to 3 seconds:

```
ACOS(config)# ip tcp syn-cookie threshold 3
```

# Configuring SYN-cookie Buffering

The following topics are covered:



## Details

When SYN cookies are enabled, 10 buffers are available to hold overflow packets from each client session. When the system memory is occupied, the number of buffers dedicated to each TCP connection is reduced. The reduction process occurs gradually and is tied to system memory usage.

There are three different thresholds that can be configured on the ACOS device. When these free system memory thresholds are breached, the number of buffers that are allocated to each session (and the TCP window size) are reduced. This reduction in the TCP window sized is an attempt to prevent the client from sending data faster than the ACOS device can receive it.

The graduated buffers and window sizes appear below. By default, each TCP session is allocated 10 buffers, and the TCP window size is set to 8K.

- If the first threshold is breached, the buffer is reduced to 4 buffers, and the TCP window size is reduced to 4K.
- If the next memory threshold is breached, the buffer is reduced to 2 buffers, and the TCP window size is reduced to 2K.
- If the final threshold is breached, the buffer is reduced to 1 buffer, and the TCP window size is reduced to 1K.

These thresholds are based on system memory usage, and the values are configurable.

Consider the following information:

- Each buffer size is approximately 1500 bytes.

The total number of buffers varies from one model to the next and is based on the total memory per connection.

- If hardware-based SYN cookies are enabled, ACOS does not modify the TCP window size.

It remains hard-coded at 65K.

### Using the GUI to Configure SYN-cookie Buffering

To configure SYN-cookie buffering using the GUI:

1. Navigate to the **ADC >> SLB >> Global** page.
2. Click the **Buffer Threshold** checkbox.

This reveals additional fields that can be configured.

**NOTE:** For more information, see the latest version of the *Online Help* for additional information about the fields.

### Using the CLI to Configure SYN-cookie Buffering

You can enter the **`buff-thresh`** CLI command to configure the thresholds for system memory usage. These threshold configurations apply to both software- and hardware-based models.

You do not have to change the system memory usage thresholds from the default settings. However, you can modify these thresholds by entering the following CLI commands:

```
!
slb common
  buff-thresh hw-buff num relieve-thresh num sys-buff-low num sys-buf-high
num
```

For additional information about changing the system memory thresholds, see the **`buff-thresh`** command in the *Command Line Interface Reference*.

## Viewing SYN-cookie Statistics

This section describes how to view SYN-cookie statistics by using the GUI or CLI.

The following topics are covered:



## Using the GUI to View SYN-cookie Statistics

To display SYN-cookie statistics, navigate to the **ADC >> Statistics >> L4** page in the GUI.

**NOTE:** For more information, see the latest version of the *Online Help* for additional information about the fields.

## Using the CLI to View SYN-cookie Statistics

This section summarizes some of the CLI commands that can be used to view SYN-cookie statistics.

The following topics are covered:



The following fields in the output of the **`show slb l4`** command allow you to view TCP traffic in terms of legitimate traffic and attacks.

### L4 SYN attack

Displays a running counter of the number of packets that the ACOS device considers to be from a SYN flood attack. This assumption is based on the fact that the device

did not receive an ACK from the client.

## L4 TCP Established

Displays a running counter of TCP packets that the ACOS device considers to be from legitimate clients. When SYN cookies are enabled, and a legitimate client sends a SYN request, the ACOS device responds with a SYN ACK. If the ACOS device receives an ACK, the packet is considered safe.

## Examples

These fields are highlighted using these examples.

The following topics are covered:

## CLI Example 1: View Attack Prevention Statistics

You can view SYN-cookies statistics for one sampling interval or across the following time intervals:

- Current
- 1 second
- 5 seconds
- 30 seconds
- 1 minute
- 5 minutes

The following command displays SYN-cookie statistics across multiple time intervals:

```
ACOS# show slb attack-prevention
                          Current    1 sec      5 sec      30 sec     1 min
     5 min
-------------------------------------------------------------------------------
---
SYN cookie snt            0          0          0          0          0
     0
SYN cookie snt ts         0          0          0          0          0
     0
```

```
SYN cookie snt fail      0          0          0          0          0
     0
SYN cookie chk fail      0          0          0          0          0
     0
SYN attack               0          0          0          0          0
     0
```

The show slb attack-prevention fields displays the fields that appear in the CLI output of the **`show slb attack-prevention`** command.

show slb attack-prevention fields

| Field | Description |
|---|---|
| SYN cookie snt | Number of TCP SYN cookies sent. |
| SYN cookie snt ts | Number of expanded TCP SYN cookies sent. |
| SYN cookie snt fail | Number of TCP SYN cookie send attempts that failed. |
| SYN cookie chk fail | Number of TCP SYN cookies for which the responding ACK failed the SYN cookie check. |
| SYN attack | Total number of SYN connections that did not receive an ACK from the client and assumed to be SYN attack. |

## Limitations

- When running the **`show slb attack-prevention`** command on an FTA model, the SYN attack field does not display output for the historical counters (1s/5s/30s/1min/5min). Output is only provided for the Current column.
- This feature is supported for L3V private partitions in non-FTA models. If the **`show slb attack-prevention`** command is run from an L3V network partitions on an FTA model, the SYN attack counter displays zero for all columns.

**NOTE:** To clear these statistics, enter the **`clear slb attack-prevention`** command.

## CLI Example 2: View SYN Attack Counter

The following example displays output from the **show slb l4** command. The L4 SYN attack field indicates that 30 packets appear to have been part of a SYN flood attack.

```
ACOS# show slb l4
                          Total
-----------------------------------------------------------------
IP out noroute            0
TCP out RST               0
TCP out RST no SYN        0
...
L4 SYN attack             30
...
```

## CLI Example 3: View Legitimate Session Counter

The following example displays output from the **show slb l4** command. The L4 TCP Established field indicates that 1,766 packets appear to have been from a legitimate source, not from an attacker.

```
ACOS# show slb l4
                          Total
-----------------------------------------------------------------
IP out noroute            0
TCP out RST               0
TCP out RST no SYN        0
...
L4 TCP Established        1766
```

## CLI Example 4: View SYN-cookie Buffering Statistics

The following example displays output for SYN cookie buffer statistics:

```
ACOS# show slb syn-cookie-buffer
Maximum SYN cookie buffer size  : 10
Total SYN cookie buffer queued  : 0
```

```
Total SYN cookie buffer drop    : 0
```

## SYN Attack Counter Support for L3V

The SYN flood attack counter in the output for the **`show slb l4`** command may not work correctly in every situation. For example, while counters that are associated with software-based SYN cookies work correctly in L3V and non-L3V deployments, counters that are associated with hardware-based SYN cookies do not work with private partitions.

The SYN flood attack counter matrix shows the limitations that are associated with using SYN flood attack counters under a variety of conditions.

SYN flood attack counter matrix

| Hardware-based SYN cookie | Software-based SYN cookie | L3V Private Partitions | SYN cookie counter incremented? |
|---|---|---|---|
| Enabled | Disabled | Disabled | Yes |
| Disabled | Enabled | Disabled | Yes |
| Disabled | Enabled | Enabled | Yes |
| Enabled | Enabled (irrelevant)[1] | Enabled | No[2] |

The SYN cookie counter incremented? column indicates whether the SYN cookie counter display will function correctly, based on the status of the other conditions that are associated with this deployment.

[1] If hardware-based and software-based SYN cookies are enabled, only hardware-based SYN cookies are used. “Irrelevant” means that hardware-based SYN cookies are also enabled.

[2] “No” means that the SYN flood attack counters fail when hardware- and software-based SYN cookies are enabled at the same time as L3V (private partitions). This is a known limitation with this feature.

# IP Limiting

IP limiting provides a an enhanced implementation of the source IP connection limiting and connection-rate limiting feature. This chapter describes the IP limiting options and how to configure and apply these options.

The following topics are covered:

# Overview of IP Limiting

IP limiting provides the following benefits:

- **Configuration flexibility:**

You can apply source IP limiting on a system-wide basis, on individual virtual servers, or on individual virtual ports.

- **Class lists:**

You can configure different classes of clients, and apply a separate set of IP limits to each class. You also can exempt specific clients from being limited.

**NOTE:** For more information, see Understanding Class Lists.

Separate limits can be configured for each of the following items:

- Concurrent connections
- Connection rate
- Concurrent Layer 7 requests
- Layer 7 request rate

**NOTE:** Layer 7 request limiting applies only to the HTTP, HTTPS, and fast-HTTP virtual port types.

# Understanding Class Lists

A class list is a set of IP host or subnet addresses that are mapped to IP limiting rules. The ACOS device can support up to 255 class lists, and each class list can contain up to 8 million host IP addresses and 64,000 subnets.

**NOTE:** Class lists can be configured only in the shared partition. A policy template that is configured in a shared partition or in a private partition can use a class list that is configured in the shared partition.

The following topics are covered:



## Class List Syntax

Each row in the class list defines a client class and has the following format:

```
ipaddr /network-mask [glid num | lid num] [age minutes] [; comment-string]
```

The

Class List Syntax Parameters

| Parameter | Description |
|---|---|
| *ipaddr* | Specifies the host or subnet address of the client. Both IPv4 and IPv6 addresses are supported. |
| *network-mask* | Subnet mask for the client address.<br><br>To configure a wildcard IP address, specify `0.0.0.0 /0` (for IPv4) or `::/0` (for IPv6). The wildcard address matches on all addresses that do not match any entry in the class list. |
| **glid** *num* | Specifies the ID of the IP limiting rule that will be used to match clients. A **glid** configures an IP limiting rule that is configured at the global configuration level. |
| **lid** *num* | Specifies the ID of the IP limiting rule that will be used to match clients. A **lid** configures an IP limiting rule that is configured at the same level as the class list (in the same policy template). |
| **age** *minutes* | Removes a host entry from the class list after the specified number of minutes. You can specify 1-2000 minutes.<br><br>When you assign an age value, the host entry remains in the class list only for the specified number of minutes. After the age reaches 0, the host entry is removed from the class list in the |

Class List Syntax Parameters

| Parameter | Description |
|---|---|
| | next minute.<br><br>You can use the age option with IP limiting options in the LID or GLID to temporarily control client access. Traffic limiting settings in the LID or GLID that are assigned to the host entry are in effect only until the age expires.<br><br>The age option applies only to host entries (IPv4 /32 or IPv6 /128). The age option is not supported for subnet entries.<br><br>**NOTE:** If you use a class-list file that is periodically re-imported, the age for class-list entries that are added to the system from the file do not reset when the class-list file is re-imported. Instead, the entries are allowed to continue aging normally. |
| *;comment-string* | Custom comment. Use a semi-colon (;) in front of the comment string.<br><br>**NOTE:** The ACOS device discards the comment string when you save the class list. |

Class List Syntax Parameters provides a description of each portion of the format.

## IP Address Matching

By default, the ACOS device matches the class-list entries based on the source IP address of client traffic. Optionally, you can also match based on one of the following items:

- **Destination IP address:**

Matches based on the destination IP address instead of the source IP address.

- **IP address in HTTP request:**

Matches based on the IP address in a header in the HTTP request. You can specify the header when you enable this option.

## Example Class Lists

Here is an example of a simple class list. This list matches on all clients and uses an IP limiting rule that is configured at the global configuration level:

```
0.0.0.0/0 glid 1
```

The following is an example with more options:

```
1.1.1.1 /32 lid 1
2.2.2.0 /24 lid 2 ; LID 2 applies to every single IP of this subnet
0.0.0.0 /0 lid 10 ; LID 10 applied to every undefined single IP
3.3.3.3 /32 glid 3 ; Use global LID 3
4.4.4.4 /32 ; No LID is applied (exception list)
```

The rows in the list specify the following:

- For individual host 1.1.1.1, use IP limiting rule 1, which is configured in a policy template.
- A policy template can be applied globally for system-wide IP limiting or to an individual virtual server or virtual port. This is described in more detail in a later section.
- For all hosts in subnet 2.2.2.0/24, use IP limiting rule 2, which is configured in a policy template.
- For all hosts that do not match another entry in the class list, use IP limiting rule 10, which is configured in a policy template.
- For individual host 3.3.3.3, use IP limiting rule 3, which is configured at the global configuration level.
- For individual host 4.4.4.4, do not use an IP limiting rule.

## Configuring Class Lists

The following topics are covered:

## Using the GUI to Import a Class List

To import a class list using the GUI:

1. Hover over ADC and select SLB from the menu bar.
2. Click the Class Lists tab, then select **Import** from the drop-down list.
3. Click **Import**.
4. Specify the name and location of the file you want to import. Refer to the GUI online help for this page for more information about each field.
5. Click **Import**.

## Using the GUI to Configure a Class List

To configure a class list using the GUI:

1. Hover over ADC and select SLB from the menu bar.
2. Click the Class Lists tab, then select **Configuration** from the drop-down list.
3. Click **Create**.
4. In the Name field, specify a class list name.
5. Complete the fields on this page as desired. Refer to the GUI online help for this page for more information about each field.

   **NOTE:** If the class list contains at least 100 entries, you should use the Store as a file option. A class list can be exported only if you use this option.

6. Click **Create**.

## Using the CLI to Import a Class List

To import a class list using the CLI, use the **`import`** command. For example:

```
ACOS(config)# import class-list vs_list ftp:
```

```
Address or name of remote host []? 1.1.1.2
User name []? ACOSadmin
Password []? *********
File name [/]? vs_list
```

### Using the CLI to Configure a Class List

To configure a class list in the CLI, use the **class-list** command. For example:

```
ACOS(config)# class-list examplelist
ACOS(config-class list)# 1.1.1.1 /32 glid 1
ACOS(config-class list)# 2.2.2.2 /32 glid 2
ACOS(config-class list)# 10.1.2.1 /32 lid 1
ACOS(config-class list)# 10.1.2.2 /32 lid 2
```

**NOTE:** See Class List Syntax for more information about the syntax.

## Understanding IP Limiting Rules

The following topics are covered:

## Parameters

IP limiting rules specify connection and request limits for clients.

Each IP limiting rule has the following parameters:

- **Limit ID** – Number from 1-31 that identifies the rule.
- **Connection limit** – Maximum number of concurrent connections that are allowed for a client. You can specify 0-1048575. Connection limit 0 immediately locks down matching clients, and there is no default value.
- **Connection-rate limit** – Maximum number of new connections that are allowed for a client in the limit period. You can specify 1-2147483647 connections. The limit period can be 100-6553500 milliseconds (ms), specified in increments of 100 ms. There is no default.
- **Request limit** – Maximum number of concurrent Layer 7 requests that are allowed for a client. You can specify 1-1048575, and there is no default.
- **Request-rate limit** – Maximum number of Layer 7 requests that are allowed for a client in the limit period. You can specify 1-4294967295 connections. The limit period can be 100-6553500 milliseconds (ms), specified in increments of 100 ms. There is no default.
- **Over-limit action** – Action to take when a client exceeds at least one limit.
- The action can be one of the following:
- **Drop** – The ACOS device drops that traffic. If logging is enabled, the ACOS device also generates a log message. This is the default action.
- **Forward** – The ACOS device forwards the traffic. If logging is enabled, the ACOS device also generates a log message.
- **Reset** – For TCP, the ACOS device sends a TCP RST to the client. If logging is enabled, the ACOS device also generates a log message.
- **Lockout period** – Number of minutes during which to apply the over-limit action after the client exceeds a limit. The lockout period is activated when a client exceeds a limit. The lockout period can be 1-1023 minutes, and there is no default.
- **Logging** – Generates log messages when clients exceed a limit. Logging is disabled by default.

When you enable logging, by default, a separate message is generated for each over-limit occurrence. If you specify a logging period, the ACOS device keeps the repeated messages for the specified period and sends a message at the end of the period for all instances that occurred during this period.

The logging period can be 0-255 minutes. The default is 0, which means that there is no wait period.

**NOTE:** When configured in a policy template, the class-list options request limit and request-rate limit are applicable only in policy templates that are bound to virtual ports. These options are not applicable in policy templates that are bound to virtual servers or in policy templates that are used for system-wide PBSLB.

**NOTE:** For more information, see Request Limiting and Request-Rate Limiting in Class Lists. The request limit and request-rate limit options apply only to HTTP, fast-HTTP, and HTTPS virtual ports. The over-limit logging, when used with the request-limit or request-rate-limit option, always lists Ethernet port 1 as the interface.

## Match IP Address

By default, the ACOS device matches class-list entries based on the source IP address of client traffic. Optionally, you can also match based on one of the following options:

- **Destination IP address** – Matches based on the destination IP address in packets from clients.
- **IP address in client packet header** – Matches based on the IP address in the specified header in packets from clients. If you do not specify a header name, this option uses the IP address in the X-Forwarded-For header.

## Request Limiting and Request-Rate Limiting in Class Lists

If a LID or GLID in a class list contains settings for request limiting or request-rate limiting, the settings apply only if the following conditions are true:

- The LID or GLID is used in a policy template.
- The policy template is bound to a virtual port.

  The settings apply only to the virtual port but do not apply in the following cases:

- The policy template is applied to the virtual server, instead of the virtual port.
- The settings are in a system-wide GLID.
- The settings are in a system-wide policy template.

**NOTE:** This limitation does not apply to connection limiting or connection-rate limiting. Those settings are valid in the cases listed above.

## CLI Examples: Request Limiting and Request-rate Limiting Settings Are Used

The following topics are covered:



### Example 1: GLID Used in Policy Template and Bound to Virtual Port

The following configuration is valid for request limiting and request-rate limiting. These settings are in a GLID that is used by a policy template that is bound to a virtual port.

```
ACOS(config)# class-list 2
ACOS(config-class list)# 5.1.1.100/32 glid 1023
ACOS(config-class list)# 55.1.1.0/24 lid 31
ACOS(config-class list)# exit
ACOS(config)# glid 1023
ACOS(config-glid:1023)# request-limit 10
ACOS(config-glid:1023)# request-rate-limit 2 per 100
ACOS(config-glid:1023)# over-limit-action reset log
ACOS(config-glid:1023)# exit
ACOS(config)# slb template policy global_policy
ACOS(config-policy)# class-list 2
```

```
ACOS(config-policy-class-list:2)# exit
ACOS(config-policy)# exit
ACOS(config)# slb virtual-server vs-55 55.1.1.55
ACOS(config-slb vserver)# vrid 1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group vlan-80-grp
ACOS(config-slb vserver-vport)# template policy global_policy
```

## Example 2: LID Used in Policy Template and Bound to Virtual Port

The following configuration also is valid for request limiting and request-rate limiting. These settings are in a LID that is configured in a policy template that is bound to a virtual port.

```
ACOS(config)# class-list l2
ACOS(config-class list)# 55.1.1.100/32 lid 31
ACOS(config-class list)# exit
ACOS(config)# slb template policy poltemplate1
ACOS(config-policy)# class-list l2
ACOS(config-policy-class-list:l2)# exit
ACOS(config-policy)# class-list l3
ACOS(config-policy-class-list:l3)# lid 30
ACOS(config-policy-class-list:l3-lid:30)# request-limit 10
ACOS(config-policy-class-list:l3-lid:30)# request-rate-limit 2 per 100
ACOS(config-policy-class-list:l3-lid:30)# exit
ACOS(config-policy-class-list:l3)# exit
ACOS(config-policy)# exit
ACOS(config)# slb virtual-server vs-55 55.1.1.55
ACOS(config-slb vserver)# vrid 1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group vlan-80-grp
ACOS(config-slb vserver-vport)# template policy poltemplate1
```

# CLI Examples: Request Limiting and Request-rate Limiting Settings Are Not Used

The following topics are covered:



## Example 1: Policy Template Bound to Virtual Server Instead of Virtual Port

The following configuration is not valid for request limiting and request-rate limiting. The policy template is bound to the virtual server instead of the virtual port.

```
ACOS(config)# slb virtual-server vs-55 55.1.1.55
ACOS(config-slb vserver)# vrid 1
ACOS(config-slb vserver)# template policy gg
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# service-group vlan-80-grp
```

## Example 2: System GLID

The following configuration is not valid for request limiting and request-rate limiting, because the settings are in a system GLID.

```
ACOS(config)# system glid 1023
```

## Example 3: System-wide Policy Template

The following configuration is not valid for request limiting and request-rate limiting, because the settings are in a policy template used for system-wide PBSLB.

```
ACOS(config)# system template policy pol1
```

## Configuring Source IP Limiting

To configure source IP limiting:

1. Configure a class list on the ACOS device or another device.
2. If you configure the class list on another device, import it to the ACOS device.
   a. Configure the following IP limiting rules:
   b. For system-wide IP limiting, configure the rules in a policy template or in standalone IP limiting rules.
3. For IP limiting on an individual virtual server or virtual port, configure the rules in a policy template.
4. Apply the IP limiting rules.

You can configure multiple policy templates with different IP limiting rules. You can use a given class list in one or more policy templates.

- For system-wide source IP limiting, apply the policy template globally.
- For source IP limiting on an individual virtual server or virtual port, apply the policy template to the virtual server or virtual port.

Clients must comply with all IP limiting rules that are applicable to the client. For example, if you configure system-wide IP limiting and also configure IP limiting on a virtual server, clients must comply with the system-wide IP limits and with the IP limits that are applied to the individual virtual server accessed by the client.

# CLI Examples - Configuration

The examples in this section show how to configure IP limiting.

The following topics are covered:

## Configuring System-wide IP Limiting With a Single Class

The following commands configure a standalone IP limiting rule to be applied globally to all IP clients, which match class list “global”:

```
ACOS(config)# glid 1
ACOS(config-glid:1)# conn-rate-limit 10000 per 1
ACOS(config-glid:1)# conn-limit 1000000
ACOS(config-glid:1)# over-limit-action forward log
ACOS(config-glid:1)# exit
ACOS(config)# system glid 1
```

The following commands configure class list “global”, which matches on all clients and uses IP limiting rule 1:

```
ACOS(config)# class-list global
ACOS(config-class list)# 0.0.0.0/0 glid 1
ACOS(config-class list)# exit
```

## Configuring System-wide IP Limiting With Multiple Classes

The commands in this example configure system-wide IP limiting by using a policy template.

```
ACOS(config)# slb template policy global_policy
ACOS(config-policy)# class-list global
ACOS(config-policy-class-list:global)# lid 1
ACOS(config-policy-class-list:global-lid...)# conn-rate-limit 20000 per 1
ACOS(config-policy-class-list:global-lid...)# conn-limit 5000000
ACOS(config-policy-class-list:global-lid...)# over-limit reset logging
ACOS(config-policy-class-list:global-lid...)# exit
ACOS(config-policy-class-list:global)# exit
ACOS(config-policy)# exit
```

The following command imports the class list that are used by the policy:

```
ACOS(config)# import class-list global_list ftp:
Address or name of remote host []? 1.1.1.2
User name []? ACOSadmin
Password []? *********
File name [/]? global_list
```

The following command applies the policy to the system:

```
ACOS(config)# system template policy global_policy
```

## Configuring IP Limiting on a Virtual Server

The commands in this example configure IP limiting for a virtual server.

The following commands configure a policy template:

```
ACOS(config)# slb template policy vs_policy
ACOS(config-policy)# class-list vs_list
ACOS(config-policy-class-list:vs_list)# lid 1
ACOS(config-policy-class-list:vs_list-lid...)# conn-rate-limit 200 per 1
ACOS(config-policy-class-list:vs_list-lid...)# conn-limit 50000
ACOS(config-policy-class-list:vs_list-lid...)# over-limit lockout 10
logging
ACOS(config-policy-class-list:vs_list-lid...)# exit
ACOS(config-policy-class-list:vs_list)# exit
ACOS(config-policy)# exit
```

The following command imports the class list that is used by the policy:

```
ACOS(config)# import class-list vs_list ftp:
Address or name of remote host []? 1.1.1.2
User name []? ACOSadmin
Password []? *********
File name [/]? vs_list
```

The following commands apply the policy to a virtual server:

```
ACOS(config)# slb virtual server vs1
ACOS(config-slb vserver)# template policy vs_policy
```

## Configuring IP Limiting on a Virtual Port

The commands in this example configure IP limiting for a virtual port.

**NOTE:** In this example, IP limiting is applied to a virtual port on a virtual server that also has IP limiting. Clients must conform to both sets of limits.

The following commands configure a policy template:

```
ACOS(config)# slb template policy vp_policy
ACOS(config-policy)# class-list vp_list
ACOS(config-policy-class-list:vp_list)# lid 1
ACOS(config-policy-class-list:vp_list-lid...)# request-rate-limit 50 per 1
ACOS(config-policy-class-list:vp_list-lid...)# request-limit 60000
ACOS(config-policy-class-list:vp_list-lid...)# over-limit reset logging
ACOS(config-policy-class-list:vp_list-lid...)# exit
ACOS(config-policy-class-list:vp_list)# exit
ACOS(config-policy)# exit
```

The following command imports the class list that is used by the policy:

```
ACOS(config)# import class-list vp_list ftp:
Address or name of remote host []?1.1.1.2
User name []? ACOSadmin
Password []? *********
File name [/]? vp_list
```

The following commands apply the policy to a virtual port:

```
ACOS(config)# slb virtual server vs1
```

```
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy vp_policy
```

## Configuring Class List Entries That Age Out

The following commands configure a class list with 2 host entries, and assign an age value to each entry.

```
ACOS(config)# class-list local
ACOS(config-class list)# 192.168.1.100 /32 lid 30 age 1
ACOS(config-class list)# 192.168.1.101 /32 lid 30 age 10
ACOS(config-class list)# exit
```

The following commands configure a policy template.

The template includes an LID that sets the connection limit to 0. The LID also resets and logs connection attempts.

```
ACOS(config)# slb template policy 1
ACOS(config-policy)# class-list local
ACOS(config-policy-class-list:local)# lid 30
ACOS(config-policy-class-list:local-lid...)# conn-limit 0
ACOS(config-policy-class-list:local-lid...)# over-limit-action reset log
ACOS(config-policy-class-list:local-lid...)# exit
ACOS(config-policy-class-list:local)# exit
ACOS(config-policy)# exit
```

The following commands apply the policy template to a virtual port.

```
ACOS(config)# slb virtual-server vs1 192.168.1.33
ACOS(config-slb vserver)# port 8080 http
ACOS(config-slb vserver-vport)# template policy 1
```

In the configuration above, host 192.168.1.100 is not allowed to establish a connection during the first minute after the host entry is created. After the age

expires, the host entry is removed form the class list, and the connection limit no longer applies to the client.

Host 192.168.1.101 is not allowed to establish a connection during the first 10 minutes after that host entry is created. Once the age expires, the client is no longer locked down.

# CLI Examples - Display

The following topics are covered:



## Viewing Class-Lists

Use the `show class-list` command to view information about your class list configuration.

**NOTE:** For information, see "**`show class-list`**" in the *Command Line Interface Reference*.

## Viewing IP Limiting Rules

Use the `show glid` command to view the configuration of each standalone IP limiting rule.

**NOTE:** For information, see "**`show glid`**" in the *Command Line Interface Reference*.

## Viewing IP Limiting Statistics

Use the `show pbslb` command to view IP limiting statistics.

**NOTE:** For information, see "**`show pbslb`**" in the *Command Line Interface Reference*.

# ICMP Rate Limiting

The following topics are covered:

# ICMP Rate Limiting Overview

ICMP/ICMPv6[1] rate limiting protects against denial-of-service (DoS) attacks such as Smurf attacks, which consist of floods of spoofed broadcast ping messages.

ICMP rate limiting monitors the rate of ICMP traffic and drops ICMP packets when the configured thresholds are exceeded.

# Configuring ICMP Rate Limiting

You can configure ICMP rate limiting filters globally, on individual Ethernet interfaces, and in virtual server templates. If you configure ICMP rate limiting filters at more than one of these levels, all filters are applicable.

The following topics are covered:



## ICMP Rate Limiting Parameters

ICMP rate limiting filters consist of the following parameters:

- **Normal rate** – The ICMP normal rate is the maximum number of ICMP packets that are allowed per second.

  If the ACOS device receives more than the normal rate of ICMP packets, the excess packets are dropped until the next one-second interval begins. The normal rate can be 1-65535 packets per second.

- **Maximum rate** – The ICMP maximum rate is the maximum number of ICMP packets allowed per second before the ACOS device locks up ICMP traffic.

[1]Subsequent references use the term “ICMP rate limiting”. Unless otherwise specified, this term also applies to ICMPv6 rate limiting.

When ICMP traffic is locked up, all ICMP packets are dropped until the lockup expires. The maximum rate can be 1-65535 packets per second.

- **Lockup time** – The lockup time is the number of seconds for which the ACOS device drops all ICMP traffic, after the maximum rate is exceeded.

  The lockup time can be 1-16383 seconds.

**NOTE:** Specifying a maximum rate (lockup rate) and lockup time is optional. If you do not specify them, lockup does not occur. Log messages are generated only if the lockup option is used and lockup occurs. Otherwise, the ICMP rate-limiting counters are still incremented but log messages are not generated.

**NOTE:** The maximum rate must be larger than the normal rate.

# Using the GUI to Configure ICMP Rate Limiting

The following topics are covered:



## Configuring ICMP Rate Limiting on an Ethernet Interface

To configure ICMP rate limiting on an Ethernet interface:

1. Navigate to the **Network** >> **Interfaces** >> **LAN** page.
2. Click the **Edit** link in the Actions column for the Ethernet interface for which you want to configure ICMP rate limiting.
3. In the Update Ethernet page, select the checkbox in the ICMP Rate Limit field, then specify the desired ICMP rate limiting parameters.

**NOTE:** For descriptions of the parameters, see ICMP Rate Limiting Parameters.

## Configuring ICMP Rate Limiting in a Virtual Server Template

To configure ICMP rate limiting in a virtual server template:

**NOTE:** This option applies only to software releases that support SLB.

1. Navigate to the **ADC** >> **Templates** >> **SLB** page.
2. Click **Create**, then select **Virtual Server** from the drop-down list.
3. In the Create Virtual Server Template page, specify the desired values in the fields beginning with "**ICMP**" or "**ICMPv6**" as desired.

**NOTE:** For descriptions of the parameters, see ICMP Rate Limiting Parameters.

## Using the CLI to Configure ICMP Rate Limiting

The following example configures a virtual server template that sets ICMP rate limiting:

```
ACOS(config)# slb template virtual-server vip-tmplt
ACOS(config-vserver)# icmp-rate-limit 25000 lockup 30000 60
```

You can enter the **icmp-rate-limit** command at any of the following configuration levels:

- Global configuration level
- Configuration level for a physical or virtual Ethernet interface
- Configuration level for a virtual server template

**NOTE:** For descriptions of the parameters, see ICMP Rate Limiting Parameters.

To view ICMP rate limiting information, enter the following commands:

```
show icmp
show icmpv6
show interfaces
show slb virtual-server server-name detail
```

# HTTP Slowloris Prevention

The following topics are covered:

# Details

The ACOS includes an HTTP template option that specifies the maximum number of seconds allowed for all parts of a request header to be received. If the entire request header is not received within the specified amount of time, ACOS terminates the connection.

This option provides security against attacks such as Slowloris attacks, which attempt to consume resources on the target system by sending HTTP requests in multiple increments, and at a slow rate. The intent of this type of attack is to cause the target system to consume its buffer resources with the partially completed requests.

**NOTE:** The request-header wait time can bet set to 1-31 seconds. The default is 7 seconds.

# Using the GUI to Configure Request Header Timeout

To configure the request header timeout using the GUI:

1. Navigate to the **ADC >> Templates >> L7** page.
2. Click **Create** and select **HTTP** from the drop-down list to create a new HTTP template.
3. On the Create HTTP Template page, select the checkbox in the **Request Header Wait Time Before Abort Connection** field, then specify a timeout value in seconds (1-31, default is 7).

# Using the CLI to Configure Request Header Timeout

To change the request-header wait time in an HTTP template, use the **`req-hdr-wait-time`** command at the configuration level for the template:

```
ACOS(config)# slb template http exampletemplate
ACOS(config-http)# req-hdr-wait-time 10
```

**NOTE:** For more HTTP security options, see the W*eb Application Firewall Guide*.

# DNS Application Firewall

The following topics are covered:

# Overview of the DNS Application Firewall

The DNS Application Firewall (DAF) provides security for DNS VIPs.

The DAF examines DNS queries that are addressed to a VIP to ensure that the queries are not malformed. If a malformed DNS query is detected, the ACOS device takes one of the following actions:

**NOTE:** These actions are specified in the DNS security policy.

- Drops the query.
- Forwards the query to another service group – This option is useful if you want to quarantine and examine the malformed queries, while keeping the queries away from the DNS server.

This feature parses DNS queries based on the following RFCs:

- **RFC 1034: Domain Names** – Concepts and Facilities
- **RFC 1035: Domain Names** – Implementation and Specification
- **RFC 2671** – Extension Mechanisms for DNS (EDNS0)

# DNS Sanity Check

The DNS security performs a sanity check on DNS client requests and, if applicable, the DNS server replies.

The following topics are covered:



## Sanity Checking for Virtual-Port Type UDP

The DNS sanity checking on virtual-port type UDP is performed only for client requests.

For a DNS client request to pass the sanity check, all of the following conditions must be met:

- Flags.qr == 0 (first bit in flags)
- Flags.opcode <=5 (bits 2 to 5 in flags)
- Flags.rcode == 0 (last 4 bits in flags)
- qdcount > 0 (questions in DNS header)

## Sanity Checking for Virtual-Port Type DNS-UDP

The DNS security performs a sanity check on DNS client requests and, if applicable, the DNS server replies.

For a client request to pass the sanity check, all of the following conditions must be met:

- Flags.qr == 0 (first bit in flags)
- Flags.opcode <=5 (bits 2 to 5 in flags)
- Flags.rcode == 0 (last 4 bits in flags)
- qdcount > 0 (questions in DNS header)

For a server response (if applicable) to pass the sanity check, all of the following conditions must be met:

- Flags.qr == 1 (first bit in flags)
- Flags.opcode <=5
- Flags.rcode == 0
- qdcount > 0
- ancount > 0 (Answer count)

# Configuring DNS Security with DAF

To configure DNS security for a DNS virtual port:

1. Create a DNS template and specify the DNS security action in the template.
2. Bind the DNS template to the DNS virtual port.

The following topics are covered:



## DNS Application Firewall Setup

The following commands configure a DNS template for DNS security and bind the template to the DNS virtual port on a virtual server. The **drop** option drops malformed queries so that they are not processed by the DNS virtual port to which the template has been applied.

```
ACOS(config)# slb template dns dns-sec
ACOS(config-dns)# malformed-query drop
ACOS(config-dns)# exit
```

The following commands configure the real server and service group:

```
ACOS(config)# slb server dns-sec1 10.10.10.88
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
ACOS(config)# slb service-group dns-sec-grp udp
ACOS(config-slb svc group)# member dns-sec1 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

The following commands bind the service group and DNS template to the DNS virtual port on a virtual server:

```
ACOS(config)# slb virtual-server dnsvip1 192.168.1.53
ACOS(config-slb vserver)# port 53 udp
ACOS(config-slb vserver-vport)# service-group dns-sec-grp
ACOS(config-slb vserver-vport)# template dns dns-sec
```

Since the **drop** action is specified, malformed DNS queries sent to the virtual DNS server are dropped by the ACOS device.

## Service-Group Redirection for DNS “Any” Requests (using aFleX)

The following aFleX script can be applied to a DNS virtual port to detect DNS “any” requests and redirect them to an alternate service group. In this example, DNS requests of type “ANY” are sent to service group `rate_limited_service_group`. DNS requests of other types are sent to service group `no_rate_limit_service_group`.

```
when DNS_REQUEST {
  set record ANY
  if {[DNS::question type] equals $record} {
    pool rate_limited_service_group
  } else {
    pool no_rate_limit_service_group
  }
}
```

# Configuring DNS Firewall Using RPZ

Response Policy Zone (RPZ) is a DNS-based security mechanism that allows you to redirect or block DNS queries based on policies defined in the RPZ file. A DNS Firewall utilizing RPZ is a powerful tool for enhancing DNS security, as it inspects and filters DNS traffic to protect against various cyber threats.

Following are the applications of a DNS Firewall:

- To block malicious domains that host malware and other malicious content, thereby protecting users from cyber threats.
- To filter content and enforce policies by blocking access to inappropriate websites.
- To mitigate phishing attacks by blocking known phishing sites.
- To redirect users to different pages in case of policy violations.

The following topics are covered:

# Implementing DNS Firewall Using RPZ

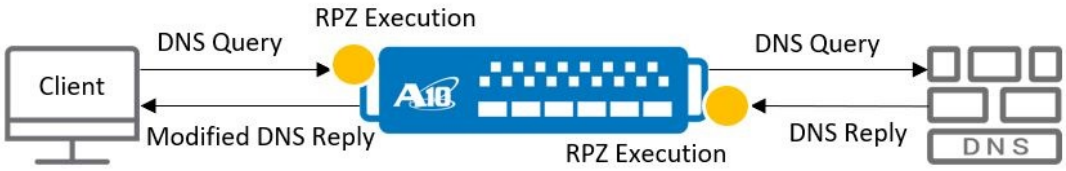
Using RPZ, you can create and distribute DNS Firewall rules within DNS zones. Each rule in an RPZ rule set is stored in a DNS Resource Record set (RRset) and consists of policy triggers and policy actions. Policy triggers are the conditions for matching DNS queries, while policy actions specify the action to be taken when a match occurs. For more information, see Policy Triggers and Actions.

## Workflow

The following workflow outlines the steps taken by a DNS Firewall utilizing RPZ to inspect, evaluate, and respond to DNS queries:

1. When a user attempts to access a domain, the DNS query is intercepted by the DNS firewall.
2. The DNS firewall processes the query and evaluates it against the RPZ rules defined in the RPZ file.
3. Based on the matched RPZ rule, the DNS firewall applies the appropriate action (block, redirect, etc.). The action can either be an error reply, a walled garden IP, or dropped packets.
4. The modified or blocked response is sent back to the user.

Figure 4 : DNS RPZ Firewall




## Policy Triggers and Actions

Policy triggers are conditions or patterns in DNS queries that need to be matched. The following table lists the common policy triggers:

Table 2 : Policy Triggers

| *Sr.No* | *Trigger Name* | *Description* | *Example* | *RPZ Format* |
|---|---|---|---|---|
| 1. | Client IP Address<br>(RPZ-CLIENT-IP) | Matches queries by the IP addresses of the DNS Client.<br>(.rpz-client-ip) | 192.168.93.180 /24 | 24.180.93.168.192.rpz-client-ip |
| 2. | Query Name (QNAME) | Matches by the query name (specific domain name).<br>(example.com) | www.a10net-works.com | www.a10net-works.com |
| 3. | Response IP Address<br>(RPZ-IP) | Matches by an IP address present in the response.<br>(.rpz-ip) | 23.185.0.2 /32 | 32.2.0.185.23.rpz-ip |
| 4. | RPZ - NSDNAM-E | Matches queries by the name of the authoritative nameserver.<br>(.rpz-nsdname) | ns-421.awsdns-52.-com | ns-421.awsdns-52.-com.rpz-nsdname |

| *Sr.No* | *Trigger Name* | *Description* | *Example* | *RPZ Format* |
|---|---|---|---|---|
| 5. | RPZ-NSIP | Matches queries by the IP address of the authoritative nameserver.<br>(.rpz-nsip) | 208.78.70.16 /32 | 32.16.70.78.208.rpz-nsip |

Policy actions refer to the action to be taken once the policy trigger is matched. The following table lists the common actions:

Table 3 : Policy Actions

| Sr.No | Action Name | Description | RPZ Format | Example |
|---|---|---|---|---|
| 1. | NXDOMAIN | Respond with an error code of type NXDOMAIN. NXDOMAIN indicates that the domain name does not exist. | (.) | `'*.maliciousdomain.com IN CNAME .'` |
| 2. | NODATA | Respond with an error code of type NODATA. NODATA indicates that the | ( *.) | `'*.maliciousdomain.com IN CNAME *.'` |

| Sr.No | Action Name | Description | RPZ Format | Example |
|---|---|---|---|---|
| | | domain name exists, but there are no records of the requested type. | | |
| 3. | PASSTHRU | Bypass filtering and forward the DNS query. | (rpz-passthru.) | '*.badsite.com IN CNAME rpz-passthru.' |
| 4. | DROP | Drop the DNS query without sending any reply. | (rpz-drop.) | '*.suspiciousdomain.com IN CNAME rpz-drop.' |
| 5. | TCP-Only | Force to resubmit query using TCP-based DNS | (rpz-tcp-only.) | 'example.com IN CNAME rpz-tcp-only.' |
| 6. | Local Data | Replace the response with specified data (in terms of RPZ-defined RR) | (arbitrary RR types) | 'www.example.com IN A 1.1.1.1' |

In case a DNS transaction matches many trigger policies, the selected action is based on the following conditions:

| Sr.No | Condition | Action |
|---|---|---|
| 1. | Same trigger policies resulting in multiple actions. | The first trigger |

| Sr.No | Condition | Action |
|---|---|---|
| | | action is selected. |
| 2. | Multiple, different trigger policies resulting in one or multiple actions. | Action is based on the following precedence:<br>• Client IP<br>• QNAME<br>• Response IP<br>• NSDNAME<br>• NSIP |
| 3. | Same trigger policies resulting in one or multiple actions. | Action is based on the order specified by RR in the DNS reply. |

**Additional Notes**

- The RPZ is a policy for the DNS response. Therefore, for better performance, some parts of RPZ such as CLIENT-IP and QNAME triggers are executed in the DNS query.
- The maximum number of rules used per RPZ depends on the maximum number of entries in the class-list.
- A maximum of 1.5 million RPZ file entries are supported, either within a single RPZ file or distributed across multiple RPZ files, with a system-wide limit of 1.5 million entries. A single RPZ file with 1.5 million entries requires approximately 4.5GB of free memory. If the entries are distributed across multiple files, additional memory may be needed.

## Example RPZ Policy

Following is an example of an RPZ file with various policy triggers and actions:

```
$TTL 1H         ;Sets the default TTL for records in this zone to 1 hour.
$ORIGIN rpz.    ;Sets the base domain for the zone to 'rpz.'.

;SOA Record
@   IN  SOA localhost. nobody.localhost (
```

```
 2015103102     ; Serial Number
 1h             ; Refresh Interval
 15m            ; Retry Interval
 30d            ; Expiry period
 2h )           ; Minimum TTL
NS localhost.

; The DROP action for matched queries
32.184.101.20.20.rpz-client-ip    IN CNAME rpz-drop.  ; Matches client IP
20.20.101.184/32

32.2.0.185.23.rpz-ip              IN CNAME rpz-drop.  ; Matches response
IP 23.185.0.2/32

www.a10networks.com               IN CNAME rpz-drop.  ; Matches the query
name www.a10networks.com

ns-130.awsdns-16.com.rpz-nsdname  IN CNAME rpz-drop.  ; Matches the
NSDNAME ns-130.awsdns-16.com

32.229.199.251.205.rpz-nsip       IN CNAME rpz-drop.  ; Matches the NSIP
205.251.199.229/32

; TCP-Only action for all subdomains of apple.com
*.apple.com          IN CNAME rpz-tcp-only.

; NXDOMAIN action (domain does not exist) for query name www.netflix.com
www.netflix.com      IN CNAME .

; NODATA action, for the queried domain www.facebook.com
www.facebook.com     IN CNAME *.

; DROP action for IPv6 addresses
128.5.C0A8.FFFF.0.1.0.db8.2001.rpz-ip IN CNAME rpz-drop.
64.5.ZZ.1.0.db8.800.rpz-ip            IN CNAME rpz-drop.
```

## Resource Record Types Supported

ACOS supports the parsing of the following DNS Resource Record (RR) types in the RPZ files:

Table 4 : Supported Resource Record Types

| Sr. No. | RR Type | Type ID (numerical value) | Description |
|---|---|---|---|
| 1 | **A** | 1 | IPv4 address record |
| 2 | **AAAA** | 28 | IPv6 address record |
| 3 | **CNAME** | 5 | Canonical Name record |
| 4 | **NS** | 2 | Name Server record |
| 5 | **SOA** | 6 | Start Of Authority record |
| 6 | **MX** | 15 | Mail Exchange record |

In addition to the above-listed RR types, ACOS also supports parsing **TYPE**<*num*> in the RPZ file for all actions mentioned in Table 3 except local-data. This implies that ACOS can also parse, and process RR types based on their Type ID or numerical value without needing their names.

For example, **TYPE1** represents **A** record. Therefore, by specifying **TYPE1** in the RPZ file, you can create a policy to process **A** records for the specified domain. **TYPE1** and **A** can be used interchangeably in RPZ file. Similarly, for **AAAA** records, you can use **TYPE28**.

Consider the following two policies:

```
malicious-ipv4.com    IN  A        127.0.0.1
malicious-ipv4.com    IN  TYPE1    127.0.0.1
```

ACOS parses and processes both policies, resulting in the same action i.e., in both cases, when a DNS query with the domain name **malicious-ipv4.com** and **A** record type is received, ACOS will respond with the IP address 127.0.0.1.

However, as mentioned earlier, ACOS only recognizes and parses RR types listed in Table 4 and **TYPE**<*num*>. Therefore, for other RR types, you must specify **TYPE**<*num*> in the RPZ file. For example, HTTPS record are not supported, therefore, to process these records, you need to specify **TYPE65**.

Consider the following two policies:

```
domain.org   IN TYPE65 .
domain.org   IN HTTPS  .
```

The first policy will be parsed and processed by ACOS, resulting in an **NXDOMAIN** response. However, the second policy will result in a parsing error (since **HTTPS** RR type is not supported), and all valid/invalid policies below it will be ignored.

## CLI Configuration

ACOS can import customized RPZ and support all its policies. ACOS applies the RPZ policy when DNS transactions occur at the virtual port with the DNS template.

- To configure a DNS Firewall using RPZ on your system, perform the following steps:

1. Import a customized RPZ file.

   To import the RPZ file, use the **import** or **import-periodic** command in the following manner:

   ```
   ACOS(config)# import rpz A10.rpz use-mgmt-port
   scp://root@192.168.93.182/root/A10.rpz
   ```

   Additionally, to ensure a secure transaction, you can import an RPZ file using DNS zone transfer with Transaction Signature (TSIG) as shown below:

   ```
   ACOS(config)# import rpz test.rpz zone-transfer use-mgmt-port
   axfr://Krpz.com.+157+20696.key@192.168.93.182/root/rpz.com
   ```

   In the above example, **Krpz.com.+157+20696.key** is the TSIG public key file that can be imported using the **import tsig** command.

2. Bind the RPZ to the DNS template.

   ```
   ACOS(config)# slb template dns dns_template1
   ACOS(config-dns)# rpz 1 A10.rpz
   ACOS(config-dns-rpz)# logging enable
   ACOS(config-dns-rpz-logging:enable)# rpz-action drop
   ACOS(config-dns-rpz-logging:enable)# rpz-action tcp-only
   ```

   **NOTE:** You cannot bind more than 8 RPZ files on the same DNS template.

3. Bind the DNS template to the virtual ports.

   ```
   ACOS(config)# slb virtual-server v1 20.20.20.1
   ```

```
ACOS(config-slb vserver)#  port 53 dns-tcp
ACOS(config-slb vserver-vport)# service-group sg2
ACOS(config-slb vserver-vport)# template dns dns_template1
```

- To relieve RPZ from the DNS template, use the following command:

```
ACOS(config-dns)# no rpz A10.rpz
```

- To delete the RPZ file, use the following command:

```
ACOS(config)# delete rpz A10.rpz
```

## Viewing the RPZ Configuration and Statistics

To view the RPZ configuration and statistics, check the following commands:

- To view the RPZ configurations, use the following command:

```
ACOS# show rpz
Max RPZ file size: 32K
Total RPZ number: 2
Name            Syntax DNS-template
-------------------------------------------------------
A10.rpz         Check  No
ADP.rpz         Check  Bind
```

- To view the RPZ file, use the following command:

```
ACOS# show rpz A10.rpz
Name:               A10.rpz
Syntax:             Check
DNS template:       Bind
Content:
;
; BIND data file for local loopback interface
;

$TTL 1H
$ORIGIN rpz.
@   IN  SOA localhost. nobody.localhost (
2015103102
1h
```

```
15m
30d
2h )
NS localhost.

; DROP action
32.184.101.20.20.rpz-client-ip        IN CNAME rpz-drop.          ;
Client ip
32.2.0.185.23.rpz-ip                  IN CNAME rpz-drop.          ;
Response IP
www.a10networks.com                   IN CNAME rpz-drop.          ; QNAME
ns-130.awsdns-16.com.rpz-nsdname      IN CNAME rpz-drop.          ;
NSDNAME
32.229.199.251.205.rpz-nsip           IN CNAME rpz-drop.          ; NSIP

; TCP-Only action
*.apple.com                           IN CNAME rpz-tcp-only.

; PASSTHRU action
www.a10networks.com                   IN CNAME rpz-passthru.

; NXDOMAIN action
www.netflix.com                       IN CNAME .

; NODATA action
www.facebook.com                      IN CNAME *.

; IPv6 example
128.5.C0A8.FFFF.0.1.0.db8.2001.rpz-ip IN CNAME rpz-drop.
64.5.ZZ.1.0.db8.800.rpz-ip            IN CNAME rpz-drop.
```

- To view the DNS RPZ statistics, use the following command:

```
ACOS(config)# show dns statistics

DNS statistics for SLB:
-----------------------
No. of requests: 0
No. of responses: 0
No. of request retransmits: 0
```

```
No. of requests with no response: 0
No. of resource failures: 0
RPZ action drop: 0
RPZ action pass through: 0
RPZ action force switching tcp: 0
RPZ action nxdomain return: 0
RPZ action nodata return: 0
RPZ action walled garden: 0
Filter type drop: 0
Filter class drop: 0
Filter type ANY drop: 0
```

- To view the DNS RPZ statistics per service counter, use the following command:

```
ACOS(config)# show slb virtual-server v1 53 <dns-udp/dns-tcp>
application-statistics

DNS RPZ Action Drop: 0
DNS RPZ Action Pass Through: 0
DNS RPZ Action Force Switching TCP: 0
DNS RPZ Action NXDOMAIN Return: 0
DNS RPZ Action NODATA Return: 0
DNS RPZ Action Walled Garden: 0
DNS RPZ Trigger Client IP: 0
DNS RPZ Trigger Response IP: 0
DNS RPZ Trigger NS IP: 0
DNS RPZ Trigger QNAME: 0
DNS RPZ Trigger NS Domain Name: 0
```

# Configuring Connection Rate Limiting policy at per LID level

The Connection Rate Limiting policy by assigning specific class-lists mapped to the LID/GLID when the global cache is enabled. This configuration helps to:

- Mitigate potentially high query rate attacks by restricting the volume of queries forwarded to DNS servers
- Prevent reflection/amplification attacks by restricting the number of identical queries allowed per class-list

To configure connection rate limiting policy at the LID level:

```
ACOS(config)# slb common
ACOS(config-common)# global-dns-cache
ACOS(config-common-global-dns-cache)# class-list cl_dns_cache
ACOS(config-common-global-dns-cache-class…)# lid 1
ACOS(config-common-global-dns-cache-class…)# conn-rate-limit 10000 per 10
ACOS(config-common-global-dns-cache-class…)# dns cache-enable
ACOS(config-common-global-dns-cache-class…)# over-limit-actiondrop lockout
10
ACOS(config-common-global-dns-cache-class…)# user-tag dns_cache_bypass
```

For more information on the global DNS cache configuration, see *Application Delivery and Server Load Balancing Guide*.

# Configuring TLD Filtering Policy

The DNS Application Firewall (DAF) provides Top-Level Domain (TLD) filtering. TLDs are used to categorize the domain names. They are the last part of a domain name located after the "dot" (e.g., .com, .org, .net). A filtering policy can be configured to allow only DNS queries with legal TLDs and log the dropped DNS queries. The DNS queries with illegal TLDs and with no TLDs are dropped.

The following is the workflow when the TLD filtering policy and logging are configured:

- The DNS queries with whitelisted TLDs (e.g., .com) are processed by the DNS server.
- The DNS server receives the query, evaluates it, and takes appropriate action.
- The DNS queries with whitelisted TLDs are forwarded to the server.
- The DNS queries with TLDs that are not whitelisted (unauthorized TLDs) are dropped by the DNS server. However, they are logged for troubleshooting and debugging purposes.

This feature applies only to dns-udp and dns-tcp ports.

To configure the TLD filtering policy, perform the following steps:

1. Configure a case-insensitive string type class-list that contains the legal TLD.

```
ACOS(config)#class-list TLD-list string-case-insensitive
ACOS(config-class-list)#str com
```

If an existing class-list is to be imported, the sample supported format is as follows:

```
class-list "tlds" string-case-insensitive file

; String (Total: 1002)
str DEMOCRAT
str DAY
str com
str SEX
str PS
```

2. Configure the SLB DNS template to enable TLD filtering and logging.

```
ACOS(config)#slb template dns d1
ACOS(config-dns)#tld-filter-white-list TLD-list
ACOS(config-dns)#tld-filter-log-enable
```

3. Configure the real server.

```
ACOS(config)# slb server RS 192.168.1.30
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
ACOS(config-real server)# exit
```

4. Configure the service group.

```
ACOS(config)# slb service-group SG dns-udp
ACOS(config-slb svc group)# member RS 53
ACOS(config-slb svc group-member:53)# exit
ACOS(config-slb svc group)# exit
```

5. Bind the SLB DNS template to a virtual server's port.

```
ACOS(config)#slb virtual-server vs1 192.168.1.33
ACOS(config-slb vserver)#port 53 dns-udp
ACOS(config-slb vserver-vport)#template dns d1
```

**Logs**

ACOS generates SYSLOG such as the following when the DNS query is dropped due to TLD filtering:

```
486707686847545345#tcp 1.2.3.4 1 4.5.6.7 1 Type=query QueryId=1345
Opcode=QUERY HeaderFlag=RD QDCount=0 ANCount=0 NSCount=0 ARCount=0
dhost=server.example.com QueryType=A QueryClass=IN TLD Filter Drop
```

ACOS generates CEF log such as the following when the DNS query is dropped due to TLD filtering:

```
486707686847545345#proto=tcp src=1.2.3.4 spt=1 dst=4.5.6.7 dpt=1 cs1=query
cs1Label=Query cn1=1345 cn1Label=Query ID cs2=QUERY cs2Label=Opcode cs3=RD
cs3Label=Header Flag cn2=0 cn2Label=Question Count cn3=0 cn3Label=Answer
Record Count cn4=0 cn4Label=Authority Record Count cn5=0
cn5Label=Additional Record Count dhost=server.example.com cs4=A
cs4Label=Query Type cs5=IN cs5Label=Query Class reason=TLD Filter Drop
```

# Configuring Filtering Policies for FQDN Label Length and Count

The DNS Application Firewall (DAF) provides filtering based on the Fully Qualified Domain Name (FQDN) label length and count. These filtering policies help mitigate DNS-based attacks where attackers use FQDNs with excessively long or numerous labels to overwhelm the DNS server and disrupt service availability.

Configure the **`label-length-filter`** and **`label-count-filter`** commands to implement these filtering policies. By configuring appropriate thresholds and actions, you can drop or forward DNS requests based on the label length or count, thereby enhancing overall security and network performance. The dropped DNS requests can also be logged for further analysis and to adjust threshold settings.

## CLI Configuration

### Configuring FQDN Label Length Filter

To enable the FQDN label length filter, configure the following commands under SLB DNS template:

```
ACOS(config)# slb template dns temp
ACOS(config-dns)# label-length-filter
ACOS(config-dns-label-count-filter)# drop-log-enable
ACOS(config-dns-label-count-filter)# action { drop | ignore }
ACOS(config-dns-label-count-filter)# fqdn-label-length <1-63> suffix <1-5>
```

The SLB template can then be bound to the DNS virtual port.

Option description:

- **action { drop | ignore }** – Configure the action to drop or forward the DNS requests when the FQDN label length exceeds the configured threshold limit.
- **drop-log-enable** - Enable logging when the DNS request is dropped.
- **fqdn-label-length** - Set the maximum length for an individual label in an FQDN.

**NOTE:** You can specify the **fqdn-label-length** parameter up to six times within one **label-length-filter** configuration, meaning you can configure up to six different rules to check label lengths in a single FQDN label-length filter configuration.

While checking the label lengths, you can also use the **suffix** parameter to specify the number of trailing labels to be ignored in an FQDN. For example, if an FQDN has 5 labels and the suffix is set to 2, the length of the first 3 labels will be checked and the last two labels will be ignored.

## Example:

The following example demonstrates the usage of the **label-length-filter** command:

```
ACOS(config)# slb template dns <template_name>
ACOS(config-dns)# label-length-filter
ACOS(config-dns-label-count-filter)# drop-log-enable
ACOS(config-dns-label-count-filter)# action drop
ACOS(config-dns-label-count-filter)# fqdn-label-length 40 suffix 3
ACOS(config-dns-label-count-filter)# fqdn-label-length 43
```

The above configuration has two rules to check the FQDN label length. As per the first rule, after ignoring the last 3 labels, if the length of other labels is greater than

40, the DNS request be dropped. As per the second rule, if the length of any FQDN label is greater than 43, the DNS request will be dropped.

## Generated Logs

Below are examples of logs generated when a DNS request is dropped by the FQDN label-length filter with the **drop-log-enable** command configured:

**Syslog:**

```
May  6 13:24:28 vThunder a10logd: [ACOS]<6> UDP 192.168.100.10 49179
192.168.100.124 53 Type=Query QueryId=44485 Opcode=Query HeaderFlag=RD|AD
QDCount=1 ANCount=0 NSCount=0 ARCount=1 dhost=google.com QueryType=A
QueryClass=IN Label Length Filter Drop
```

**CEF example:**

```
May  6 13:24:28 vThunder CEF:0|A10|CFW|6.0.4-d-
484066f|486707618128068611|Log DNS Query Drop for FQDN label length
Filter|2|proto=UDP src=192.168.100.10 spt=49179 dst=192.168.100.124 dpt=53
cs1=Query cs1Label=Query cn1=44485 cn1Label=Query ID cs2=Query
cs2Label=Opcode cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question
Count cn3=0 cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record
Count cn5=1 cn5Label=Additional Record Count dhost=google.com cs4=A
cs4Label=Query Type cs5=IN cs5Label=Query Class reason=Label Length Filter
Drop
```

## Configuring FQDN Label Count Filter

To enable the FQDN label count filter, configure the following commands under SLB DNS template:

```
ACOS(config)# slb template dns <template_name>
ACOS(config-dns)# label-count-filter
ACOS(config-dns-label-count-filter)# drop-log-enable
ACOS(config-dns-label-count-filter)# action { drop | ignore }
ACOS(config-dns-label-count-filter)# min-fqdn-label-count <1-31>
ACOS(config-dns-label-count-filter)# max-fqdn-label-count <1-7>
```

The SLB template can then be bound to the DNS virtual port.

Option description:

- **action { drop | ignore }** – Configure the action to drop or forward the DNS requests when the FQDN label count exceeds the configured threshold limit.
- **drop-log-enable** - Enable logging when the DNS request is dropped.
- **max-fqdn-label-count** - Set the maximum number of FQDN labels allowed per FQDN. The configured action is taken when this threshold is breached. For example, if the count is set to 5, DNS requests with FQDNs having 5 or more labels are dropped.
- **min-fqdn-label-count** - Set the minimum number of FQDN labels allowed per FQDN. The configured action is taken if the number of labels is less than this value. For example, if the count is set to 2, DNS requests with FQDNs having single labels are dropped.

## Example:

The following example demonstrate the usage of the **label-count-filter** command:

```
ACOS(config)# slb template dns temp
ACOS(config-dns)# label-count-filter
ACOS(config-dns-label-count-filter)# drop-log-enable
ACOS(config-dns-label-count-filter)# action drop
ACOS(config-dns-label-count-filter)# min-fqdn-label-count 2
ACOS(config-dns-label-count-filter)# max-fqdn-label-count 6
```

As per this configuration, DNS requests with FQDNs having single labels or more than 6 labels will be dropped.

## Generated Logs

Below are examples of logs generated when a DNS request is dropped by the FQDN label-count filter with the **drop-log-enable** command configured:

**Syslog example:**

```
May  3 17:53:05 vThunder a10logd: [ACOS]<6> UDP 192.168.100.10 54315
192.168.100.124 53 Type=Query QueryId=41043 Opcode=Query HeaderFlag=RD|AD
QDCount=1 ANCount=0 NSCount=0 ARCount=1 dhost=hp.com QueryType=A
QueryClass=IN Label Count Filter Drop
```

**CEF Log example:**

```
May  3 17:53:10 vThunder CEF:0|A10|CFW|6.0.4-d-
484066f|486707618128068613|Log DNS Query Drop for FQDN label count
Filter|2|proto=UDP src=192.168.100.10 spt=54315 dst=192.168.100.124 dpt=53
cs1=Query cs1Label=Query cn1=41043 cn1Label=Query ID cs2=Query
cs2Label=Opcode cs3=RD|AD cs3Label=Header Flag cn2=1 cn2Label=Question
Count cn3=0 cn3Label=Answer Record Count cn4=0 cn4Label=Authority Record
Count cn5=1 cn5Label=Additional Record Count dhost=hp.com cs4=A
cs4Label=Query Type cs5=IN cs5Label=Query Class reason=Label Count Filter
Drop
```

# DNS Response Rate Limiting

The following topics are covered:



## Overview

For some ADC deployments, it may be difficult to control the rate of DNS responses from the DNS servers to external hosts. This vulnerability could cause your network resources to be used in DNS reflection, DNS amplification, or DNS Water Torture attacks.

To address this vulnerability, ACOS offers support for DNS Response Rate Limiting (RRL) to mitigate the risk associated with such attacks. With DNS RRL, ACOS can effectively control the rate of DNS server responses associated with the DNS requests flagged as potentially malicious.

### DNS Amplification Attacks

The DNS amplification attacks, also known as high query rate attacks, occur when an attacker floods a DNS server with excessive DNS queries within a short period. These queries are often designed in a way that they require lengthy responses, thus, amplifying the impact of the attack.

Using DNS RRL, ACOS can mitigate the potential impact of the high query rate attacks by restricting the volume of queries forwarded to DNS servers per class-list.

### DNS Reflection Attacks

A DNS reflection attack is when a hacker hijacks multiple computers using botnets and then sends a large number of queries to one or more DNS servers. The hacker's DNS requests include a spoofed source IP address, so it appears the spoofed DNS queries originate from a legitimate source (i.e., the intended victim's address). The unwitting DNS server replies to the spoofed address of the victim instead of

replying to the real source of the threat. When the hacker scales up the attack by employing botnets, the replies from the DNS servers can use up all the resources on the target's network, preventing legitimate traffic from getting through.

Using DNS RRL, ACOS can prevent the reflection attacks by restricting the number of identical queries per class-list.

## DNS Water Torture Attacks

The DNS Water Torture attacks occur when an attacker floods a DNS server with a high volume of requests for non-existent domains (NXDOMAIN). This overwhelms the server and prevents it from processing legitimate DNS queries, and thus, disrupts normal operations.

Using NXDOMAIN RRL, ACOS can mitigate the potential impact of DNS Water Torture attacks by restricting the rate of NXDOMAIN responses at:

- SLB DNS template per vPort level
- DNS RRL class-list LID

The NXDOMAIN response rate limit is enabled only when the filter-response-rate limit is exceeded.

## Challenges of Stopping DNS Amplification or Reflection Attacks

DNS runs on a connectionless UDP protocol, so it is difficult to validate each DNS query and drop malicious traffic in a targeted manner. However, ACOS employs certain identification criteria to mitigate such threats.

ACOS identifies potentially malicious DNS queries based on the following criteria:

- There are an excessive number of queries.
- They originate from the same domain.
- They request the same FQDN-to-IP mapping from the DNS server.

Once the source is flagged as potentially malicious, then ACOS can initiate protective measures.

## DNS Response Rate Limiting

RRL is applied to DNS server responses associated with queries flagged as potentially malicious. The DNS RRL, a BIND feature, reduces the unnecessary load on the authoritative DNS servers.

**NOTE:** DNS RRL is implemented based on *ISC-TN-2012-1-Draft1*, which uses both **BIND9** and **NSD**.

BIND software tracks all DNS queries by placing them into one large table. However, to allocate system resources more efficiently, the ACOS implementation of DNS RRL uses a two-tiered system with two tables.

- **filter table -** This table processes all the normal DNS queries. When the number of requests from a source address/FQDN pair exceeds the filter-table-response rate, this source address/FQDN pair is added and tracked in the rate-limiting entry table.
  - Size depends on the platform:

    On platforms with less than 7 CPUs, the size is 4096 bytes.

    On platforms >= 7 CPUs, the size is 12288 bytes.
  - Refill cycle:

    ACOS 4.x release refills every 2 seconds

    ACOS 5.x release refills every 1 second.
- **rate-limiting entry table** - This table tracks the DNS requests that are potentially malicious or abnormal, which are sent from offenders and must be closely monitored. Only a small subset of DNS queries is placed into this table of potential abusers. It uses approximately 100 bytes for each DNS query. After all the entries in the table are used up, all other traffic is placed into an overflow bucket where the source IP + FQDN is no longer tracked.

  The rate-limiting table allocates a credit rate to each source address/FQDN pair entry (for example, a credit of up to 10 requests per second), which can be used up. Any DNS queries exceeding their credit rate are then rate-limited, and ACOS drops the traffic beyond the threshold.

**NOTE:** ACOS does not apply rate limits to the malicious queries themselves but only to the responses from the DNS server to the victim. DNS RRL is not supported on service partitions.

While the RRL feature is typically used only by authoritative name servers to mitigate the impact of DDoS amplification attacks, it can also be used by recursive servers (resolvers) as a method of load limiting. However, disabling or restricting recursion in the name server is highly recommended.

If a resolver is required, it should only be available as a non-authoritative server that can only be accessed by the intended clients, preferably those with IP addresses that cannot be spoofed as, for example, private networks.

## DNS RRL Configuration Options

DNS RRL can be configured using the following options in the SLB template DNS.

**NOTE:** DNS RRL feature works when the template is bound to virtual port type **`dns-udp`** only.

The following options are available under **`slb template dns`** > **`response-rate-limiting`**:

- **TC-rate** - This option configures the rate at which the DNS server responds with a truncated (TC) response. Every nth rate limited request will get a TC bit response and force the requestor to use TCP.

  The value can be set from 2 - 10. For example, if the TC-Rate is set to 3, one of every 3 rate-limited (dropped) queries will receive a truncated response.

- **action** - This option configures the action to be taken if the DNS response rate limit exceeds. The following options can be configured:

  - **log-only** - Enables “log only” behavior for rate limiting. ACOS will behave as if the queries are being rate-limited. Logs will be sent out, and counters will increment, but this is done without actually applying rate limits to DNS responses. Enabling this option also requires selecting the “enable-log” configuration.

- **rate-limit** - Rate-Limit based on configuration (default).
- **whitelist** - Disables DNS rate-limiting.

- **filter-table (filter-response-rate)** - This table processes all the normal DNS queries. When the number of requests from a source address/FQDN pair exceeds the filter-table-response rate, this source address/FQDN pair is added and tracked in the rate-limiting entry table (second table).

  The default value is set to 10 queries per second. The value can be from 1 - 1000 queries per second.

- **rate-limiting entry table** (**response-rate**) - This is the second table that processes all the potentially malicious queries stored here, and the RRL settings are applied. This option can be configured at:

  - DNS RRL Class-List LID template level where the value can range from 0 - 16000000 responses per configured window where 0 indicates unlimited responses without restriction on the response rate.
  - SLB template per virtual port level where the value can range from 1 - 1000 responses per configured window.

  The default value is set to 5 per second.

  You can also prevent DNS attacks from consuming too much system memory in the rate-limiting entry table using the **`max-table-entries`** (under **`slb common`** > **`dns-response-rate-limiting`**) system-level option.

- **nxdomain response rate** - This option configures the maximum allowed rate of non-existent domain (NXDOMAIN) responses. The responses that exceed this value are dropped without considering truncated responses or partial allowance through slip rates.

  This option can be configured at:

  - SLB template per virtual port level where the value can range from 1 - 1000 responses per configured window.
  - DNS RRL Class-List LID template level where the value can range from 1 - 16000000 responses per configured window.

  The default value is set to 5 per second.

When the traffic exceeds the **filter-response-rate** limit, individual client tracking begins and the system applies the configured NXDOMAIN response rates as follows:

- If the client matches a configured LID (Local Identifier), the DNS RRL class-list LID Rate limit defined in the LID is applied.
- If the client does not match a configured LID, the SLB DNS template rate defined is applied.

You can optimize resource utilization during DNS rate-limiting by removing inactive source entries from the DNS rate-limiting table after a specified period. This helps release system resources and reduces the risk of resource exhaustion. To do so, use the **source-entry-age** (under **slb common** > **dns-response-rate-limiting**) partition-level option.

**NOTE:** **TC-rate** and **slip-rate** are not supported for NXDOMAIN RRL.

- **match subnet (IPv4 or IPv6)** - This option configures the IPv4 or IPv6 prefix length to indicate the size of the subnet in which the incoming queries are grouped.
- **slip rate** - This option allows a certain percentage of valid DNS queries to pass through, even during an attack. Every nth response that is rate-limited will instead be let through.

  The value can be set from 2 - 10 and should approximate the retry count for regular queries.

**NOTE:** **TC-rate** and **slip-rate** are mutually exclusive options.

- **source IP only** - This option allows response rate limiting only based on source IP instead of FQDN.
- **window** - This option configures the rate-limiting-window. It is the interval over which rates are measured for `response-rate` and `slip-rate`. If the same DNS mapping is requested too many times, similar queries from that client are dropped for the rest of the window's interval.

  The default value is 1 second, and the value can be from 1 - 60 seconds.

For more information, see *Command Line Interface Reference*.

# Configuration Example

The following topics are covered:



## CLI Configuration

### Example 1

To configure DNS RRL using SLB Template DNS, use the following commands:

1. Enable DNS RRL:

```
ACOS(config)# slb common
ACOS(config-common)# dns-response-rate-limiting
ACOS(config-common-dns-response-rate-limi...)# max-table-entries 20000
ACOS(config-common-dns-response-rate-limi...)# exit
```

2. Define SLB template DNS:

```
ACOS(config)# slb template dns DNSRRL
ACOS(config-dns)# response-rate-limiting
ACOS(config-dns-response-rate-limiting)# response-rate 5
ACOS(config-dns-response-rate-limiting)# filter-response-rate 10
ACOS(config-dns-response-rate-limiting)# slip-rate 5
ACOS(config-dns-response-rate-limiting)# enable-log
ACOS(config-dns-response-rate-limiting)# exit
```

3. Configure the real server:

```
ACOS(config)# slb server RS 10.10.10.1
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
```

4. Configure service groups:

```
ACOS(config)# slb service-group SG udp
```

```
ACOS(config-slb svc group)# member RS 53
ACOS(config-slb svc group-member:53)# exit
```

5. Configure a virtual server and a port on the ACOS device and associate them with the proper service group:

```
ACOS(config)# slb virtual-server VS 10.10.10.2
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# template dns DNSRRL
ACOS(config-slb vserver-vport)# service-group SG
ACOS(config-slb vserver-vport)# exit
```

## Example 2

To configure NXDOMAIN Response Rate Limiting, use the following commands:

1. Enable DNS Response Rate Limiting and set the source entry age:

```
ACOS(config)# slb common
ACOS(config-common)# dns-response-rate-limiting
ACOS(config-common-dns-response-rate-limi...)# source-entry-age 5
ACOS(config-common-dns-response-rate-limi...)# exit
ACOS(config-common)# exit
```

2. Configure the class-list for DNS traffic:

```
ACOS(config)# class-list a10 dns
ACOS(config-class list)# dns contains a10 lid 1
```

3. Configure NXDOMAIN Response Rate Limiting at SLB template per virtual port:

```
ACOS(config)# slb template dns nx
ACOS(config-dns-response-rate-limiting)# response-rate-limiting
ACOS(config-dns-response-rate-limiting)# nx-response-rate 1
```

4. Configure NXDOMAIN Response Rate Limiting at DNS RRL Class-List LID template level:

```
ACOS(config-dns-response-rate-limiting)# rrl-class-list a10
ACOS(config-dns-response-rate-limiting-rr...)# lid 1
ACOS(config-dns-response-rate-limiting-rr...)# nx-response-rate 3
ACOS(config-dns-response-rate-limiting-rr...)# exit
ACOS(config-dns-response-rate-limiting)# exit
```

5. Configure the real server:

```
ACOS(config)# slb server RS 10.10.10.1
ACOS(config-real server)# port 53 udp
ACOS(config-real server-node port)# exit
```

6. Configure service groups:

```
ACOS(config)# slb service-group SG udp
ACOS(config-slb svc group)# member RS 53
ACOS(config-slb svc group-member:53)# exit
```

7. Configure a virtual server and a virtual port on the ACOS device and associate them with the proper service group:

```
ACOS(config)# slb virtual-server vip 10.10.10.2
ACOS(config-slb vserver)# port 53 dns-udp
ACOS(config-slb vserver-vport)# template dns DNSRRL
ACOS(config-slb vserver-vport)# service-group SG
ACOS(config-slb vserver-vport)# exit
```

## GUI Configuration

The DNS RRL feature helps prevent network equipment (DNS authoritative servers) from becoming unwanted participants in a DNS reflection or DNS amplification attack.

To configure DNS Response Rate Limiting using SLB Template DNS:

1. Navigate to the **ADC** > **Templates** > **L7 Protocols** menu.
2. Click **Create**, and select **DNS** from the drop-down menu.
3. Select the **DNS Response Rate Limiting** checkbox.
4. You can configure the options from this page to enable DNS Response Rate Limiting (RRL).
5. Click **OK** to save your changes.

To set limits around the amount of memory consumed during a DNS reflection attack:

1. Navigate to **ADC** > **SLB** > **Global**.
2. Select the **DNS Response Rate Limiting** checkbox.
3. From the **Max Table Entries** field that appears, specify the desired value.
4. Click **Update** to save your changes.

## Show Commands

This section describes the various show commands:

- To view normal DNS traffic (without RRL settings), use the following command:

```
ACOS(config)#show dns statistics
```

```
DNS statistics for SLB:
-----------------------
No. of requests: 0
No. of responses: 0
No. of requests with no response: 0
No. of request retransmits: 0
No. of requests and responses not match: 0
No. of resource failures: 0
DNS requests drop: 0
Filter type drop: 0
Filter class drop: 0
Filter type ANY drop: 0
RPZ action drop: 0
RPZ action pass through: 0
RPZ action force switching tcp: 0
RPZ action nxdomain return: 0
RPZ action nodata return: 0
RPZ action walled garden: 0
DNS statistics for IP NAT:
--------------------------
DNS requests drop: 0
No. of requests: 0
No. of responses: 0
No. of requests with no response: 0
No. of request retransmits: 0
No. of resource failures: 0
No. of requests reusing a transaction id: 0
DNS statistics others:
--------------------------
No. of successful client tls: 0
No. of successful server tls: 0
No. of backend udp conns created: 0
No. of backend udp conns established: 0
No. of paddings to server stripped: 0
No. of paddings to client added: 0
No. of edns client subnet to server removed: 0
No. of udp retransmission: 0
```

```
No. of udp retransmission failed: 0
```

- To view the DNS RRL entries, use the following command:

```
ACOS(config)#show dns response-rate-limiting entries
Source Address          FQDN                        Hit Count
----------------------+--------------------------+----------
10.211.3.101           test4.example.com          4
10.211.3.100           test4.example.com          3
10.211.3.101           test0.example.com          4
10.211.3.100           test0.example.com          4
10.211.3.101           test1.example.com          3
10.211.3.100           test1.example.com          3
10.211.3.101           test3.example.com          3
10.211.3.100           test3.example.com          4
10.211.3.20            test2.example.com          4
10.211.3.21            test4.example.com          4
10.211.3.22            test0.example.com          3
10.211.3.23            test1.example.com          3
10.211.3.24            test3.example.com          4
10.211.3.101           test2.example.com          4
10.211.3.100           test2.example.com          3
Total Entries: 15
```

- To view the DNS RRL statistics, use the following command:

```
ACOS(config)#show dns response-rate-limiting statistics
                                    Total
-------------------------------------------------------------------
Current Entry Count                 30
Total Entry Created                 20
Total Entry Freed                   10
Total Overflow Entry Hits           0
Total Logs                          2
```

- To view the DNS RRL statistics of a virtual server, use the following command:

```
ACOS(config)# show slb virtual-server v1 53 dns-tcp application-
statistics
Total DNS Query: 0
Total Malformed Query: 0
```

```
DNS Response Rate Limiting Total Allowed: 0
DNS Response Rate Limiting Total Dropped: 0
DNS Response Rate Limiting Total Slipped: 0
DNS Response Rate Limiting Total TC: 0
DNS Response Rate Limiting Bad FQDN: 0
Total DNS Filter Query Type Drop: 0
Total DNS Filter Query Class Drop:       0
DNS Filter Query Type A Drop: 0
DNS Filter Query Type AAAA Drop: 0
DNS Filter Query Type CNAME Drop: 0
...
```

- To view the counters for NXDOMAIN response rate limiting, use the following command:

```
ACOS(config-dns-response-rate-limiting)# show slb virtual-server vip 53
dns-udp application-statistics
...
...
RRL NXDOMAIN Exceed (Server-Side):              6
RRL QPS Drop / Log (Client-Side):               0
RRL NXDOMAIN Drop/ Log (Client-Side):           1
...
...
Response Rate Limiting Total Allowed:           0
Response Rate Limiting Total Dropped:           0
```

# DNSSEC Support

This chapter describes the ACOS device's DNSSEC support.

The following topics are covered:

# Overview of DNSSEC Support

The following topics are covered:



## Details

An ACOS device that is configured as a Global Server Load Balancing (GSLB) controller can act as an authoritative DNS server for a domain zone. As the authoritative DNS server for the zone, the ACOS device sends records in response to requests from DNS clients. The ACOS device supports the ability to respond to client requests for the following types of records:

- A
- AAAA
- CNAME
- NS
- MX
- PTR
- SRV
- TXT

If you place the ACOS device in the DNS infrastructure, the device is exposed to potential online attacks. When DNS was originally designed, there were no mechanisms to ensure the DNS infrastructure would remain secure.

In an unsecured DNS environment, the client's DNS resolver has no way to assess the validity of the address it receives for a particular domain name, so the client's DNS resolver cannot tell whether an address received for a particular domain is from the legitimate owner of that domain.

This potential security hole makes DNS vulnerable to “man-in-the-middle” attacks, DNS cache poisoning attacks, and other online attacks that could be used to forge DNS data, hijack traffic, and to potentially steal sensitive information from the user.

To close this security hole, in the 1990s, the Internet Engineering Task Force (IETF) introduced a set of standards called Domain Name System Security Extensions (DNSSEC). These additional standards add authentication to DNS and help ensure the integrity of the data that is transferred between the client resolvers and DNS servers.

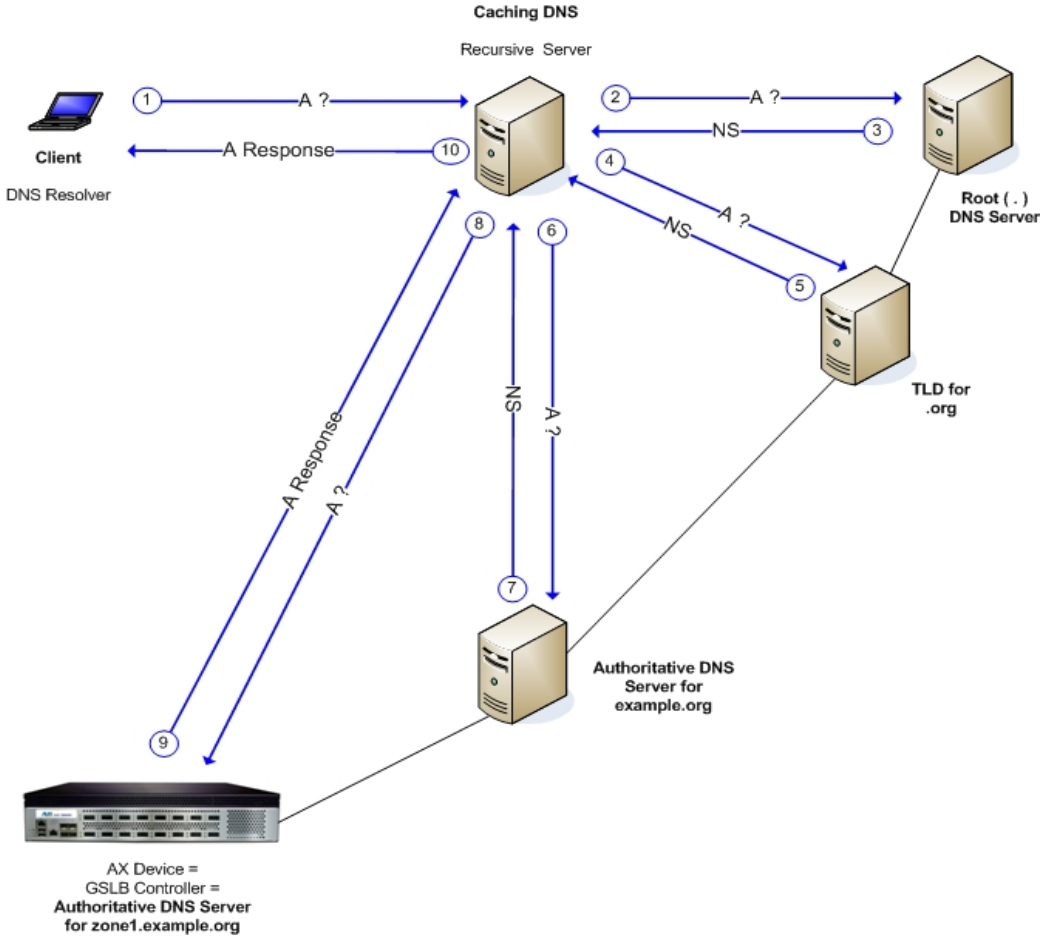
DNSSEC offers authentication through the use of cryptographic keys and digital signatures, which ensure that entries in DNS tables are correct and that connections are made to legitimate servers. The ACOS device’s implementation of DNSSEC is based on RFCs 4033, 4034, and 4035.

**NOTE:** DNSSEC for GSLB is not supported in proxy mode.

## DNS without Security

The DNS Packet Flow without DNSSEC illustrates basic DNS without DNSSEC. The figure shows the recursive lookup process that occurs when a client resolver requests the IP address for a URL. Note that this illustration shows how a client request works in a simple DNS environment without DNSSEC.

Figure 5 : DNS Packet Flow without DNSSEC




A client (shown at upper left) requires access to a server in the domain *zone1.example.org* (at lower left). The ACOS device, which is acting as the GSLB controller, is the authoritative DNS server for the zone. To access this server, the client requires the IP address for this zone or domain.

When the user enters the domain name in the web browser's URL, the process to obtain the IP address that is associated with this domain is as follows:

1. The DNS resolver that is embedded in the client's web browser sends an address request ("**A?**") to the Caching DNS server to see whether the Caching DNS server has the required IP address cached in its memory for the requested *example.org* domain.

2. The Caching DNS server has a list of IP address-to-domain mappings, but the list is not comprehensive, and unfortunately, the Caching DNS server does not have the required IP address.

   It acts as a proxy for the client and makes a recursive query to the Root DNS Server, which is located at the top of the DNS hierarchy.

3. The Root DNS Server does not have the requested IP address.
4. In an attempt to point the Caching DNS server in the right direction, it responds to the request with a Name Server (NS) record, which contains the IP of the Top Level Domain (TLD) server for the "***.org***" domain.
5. The Caching DNS server now has the IP address for the name server that manages the "***.org***" domain, so it sends an address request on behalf of the client to the TLD DNS server for the "***.org***" domain.
6. The TLD Server does not have the requested IP address.
7. The TLD server points the Caching DNS server in the right direction by providing an NS record that contains the IP address for the next name server in the DNS hierarchy, which is the authoritative DNS server for the *example.org* subdomain.
8. The Caching DNS server has the IP address that is needed to reach the authoritative DNS server for the *example.org* domain, so the server sends a request for *zone1.example.org* to this authoritative DNS server.
9. The authoritative DNS server does not have the requested information, but it can get the Caching DNS server one step closer to its destination by providing the NS record for the authoritative DNS server for the *zone1.example.org* domain.
10. The Caching DNS Server sends a request to the authoritative DNS server for the *zone1.example.org* domain.
11. The ACOS device, which is the authoritative DNS server for *zone1.example.org*, has the IP address that the client needs.
12. The ACOS device sends the requested IP address to the Caching DNS server.
13. The Caching DNS server sends the IP address that is provided by the ACOS device to the DNS resolver in the client's browser.

The client now has the IP address needed to reach the server in the zone1 subdomain.

# DNSSEC (DNS with Security)

The Figure 6 illustrates how the DNS query process works when the security extensions are used with DNS to provide security (DNSSEC). The process is similar to the process illustrated in the DNS Packet Flow without DNSSEC, but with the notable exception that DNSSEC uses the following additional resource record types to provide security:

- **DNS Key (DNSKEY)** – Public key used by an Authoritative DNS server to sign resource records for its zone.
- **Delegation Signer (DS)** – Hash (message digest) of a public key. A DNS server uses the DS for a zone that is directly underneath it in the DNS hierarchy to verify that signed resource records from the Authoritative DNS server for that zone are legitimate.
- **Resource Record Signature (RRSIG)** – Holds the digital signature of an RRSet.
- **Next Secure Record (NSEC)** and **Next Secure Record version 3 (NSEC3)** – They contain a link to the next secure record name in the zone. They are used for explicit denial-of-existence of a DNS record. For example, if you request DNS for the IP address of a domain that does not exist, the response is an empty answer. NSEC and NSEC3 are used to allow for an authenticated denial of existence. NSEC records are signed and thus can be validated through their RRSIG.

As the first step in securing a zone with DNSSEC, all the records with the same name/type/class are grouped into a resource record set (RRset). For example, in case of three AAAA records in the zone at the same label (i.e. test.a10net.com), the records are all grouped into a single AAAA RRset. This entire RRset gets digitally signed, as against individual DNS records.

The digital signature is created by applying a hash function to the DNS record to reduce its file size, an encryption algorithm is applied to the hash value (using the private key). The encrypted hash value appears as the digital signature stored in RRSIG record. It appears at the bottom of the record being signed.

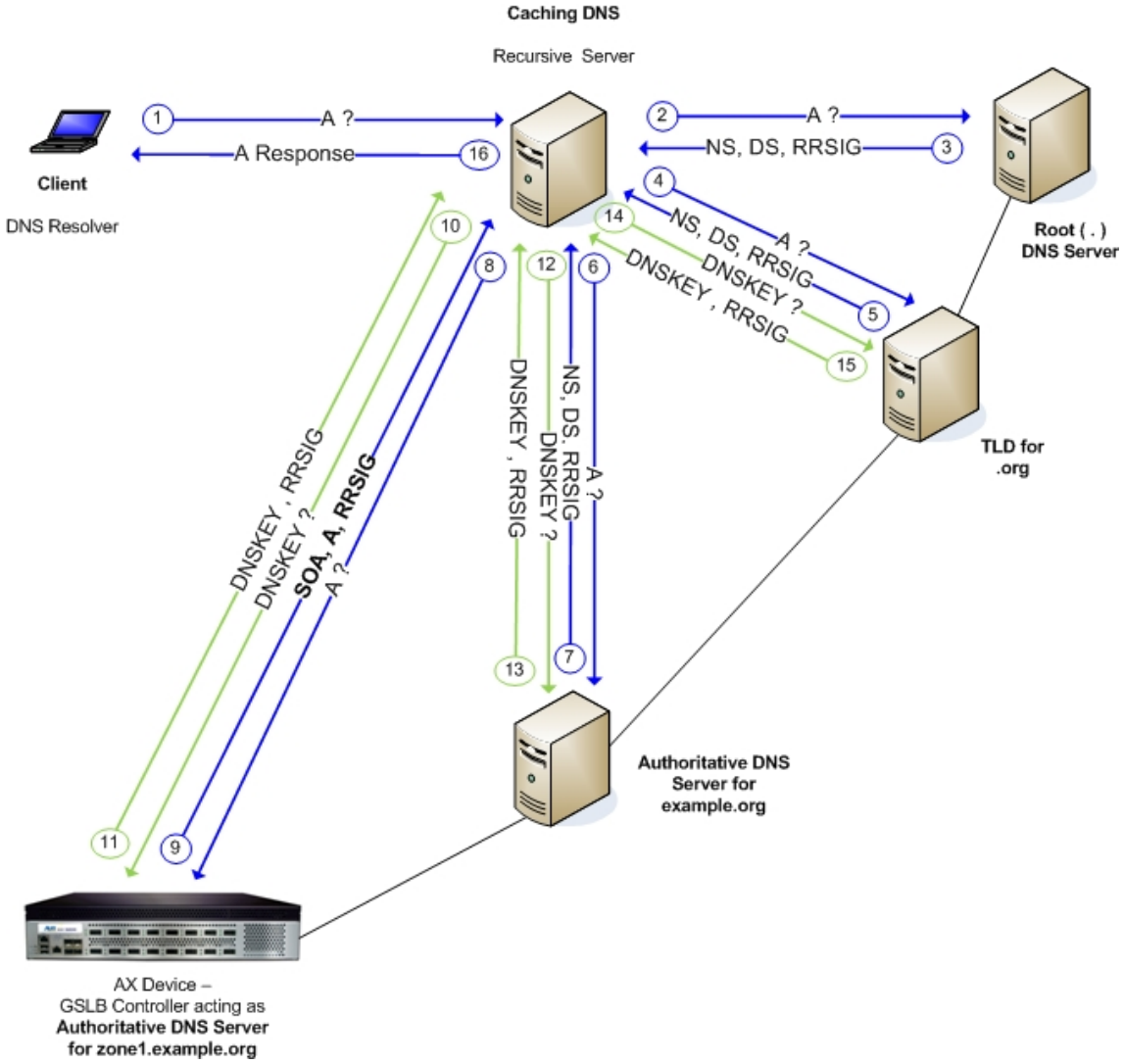
While the DNS Packet Flow without DNSSEC shows how basic DNS works without DNSSEC, the Figure 6 shows how the DNS lookup process works with DNSSEC.

The recursive lookup process remains largely unchanged, with the higher-level DNS servers pointing to lower level servers in the DNS hierarchy to move the request closer to the authoritative server for the desired domain.

However, when DNSSEC is added, the additional records such as DS, RRSIG, and DNSKE are used to sign and authenticate the communications from the DNS servers. This step proves to the client that each of the name servers in the "**chain of trust**" are authoritative for their respective domains. DNS resolver sets the "DO" (DNSSEC Ok) flag in its queries to indicate that it supports DNSSEC and servers that support DNSSEC should consider this flag.

**NOTE:** For more details, Building the Chain of Trust.

Figure 6 : DNS Packet Flow with DNSSEC




The Figure 6 shows the resolution process for an address query from the DNS resolver on a client for the IP address of *zone1.example.org*.

1. The DNS resolver on the client sends an address query for the IP address of a host under *zone1.example.org*.
2. The Caching DNS server, which does not have the address, forwards the request to the root server.
3. The root server redirects the Caching DNS server to the TLD DNS server for the *.org* domain.

This is accomplished by sending an NS record with the IP address of that TLD server. The root server uses an RRSIG record, which is used to store the private key, to sign the NS record, and the root server sends a copy of the DS record to the Caching DNS server, which points to the TLD server.

4. The Caching DNS server sends the address query to the TLD server for the .org domain.
5. The TLD server does not have the requested address, so it points the Caching DNS server to the Authoritative DNS server for *example.org*.
6. The TLD server sends an NS record with the IP address of the authoritative server for *example.org*, and the TLD server signs the NS record with the private key in the RRSIG record.
7. The Caching DNS server sends the address query to the Authoritative DNS server for *example.org*.
8. The Authoritative DNS server for *example.org* does not have the requested address, so it responds to the caching server's request by sending the NS record (signed with the RRSIG record).
9. This NS record contains the IP address of the Authoritative DNS server for *zone1.example.org*.
10. The server sends the DS record for the *zone1.example.org* server to the Caching DNS server.
11. The Caching DNS server sends the address query to the Authoritative DNS server for *zone1.example.org*, which happens to be the ACOS device.
12. The Caching DNS server has reached the Authoritative DNS server for *zone1.example.org*.
13. The Authoritative DNS server (which is the ACOS device) replies with an SOA record, the requested A record, and RRSIG records that contains the private key, which is used to sign the SOA and A records.
14. The Caching DNS server asks the ACOS device for its DNSKEY record, which is where the public key for the zone is advertised.
15. This public key is needed to unlock the resource records and verify the hash values back up the chain.

16. The ACOS device sends its DNSKEY record, with an RRSIG record that was used to sign the DNSKEY record.
17. The RRSIG record contains the private key.
18. To continue assembling the chain of trust, the Caching DNS server asks the Authoritative DNS server for *example.org* for its DNSKEY record.
19. The Authoritative DNS server for *example.org* sends its DNSKEY record with an RRSIG record (with the private key) that was used to sign the DNSKEY record.
20. The Caching DNS server asks the TLD server for *.org* for its DNSKEY record.
21. The TLD server sends its DNSKEY record with an RRSIG record that was used to sign the DNSKEY record.
22. The Caching DNS server now has all the private/public key pairs and has validated all of the links in the chain of trust.

The Caching DNS server can now send the trusted response to the DNS resolver on the client.

## DNSSEC Data and Validation

DNS query and response contain DNSSEC-related flags (bits) to indicate if DNSSEC data is included and if it was validated.

These flags are set in the DNS packet header. The following flags can be set:

- DO: The "DNSSEC OK" or DO bit set in a DNS query indicates that the client is DNSSEC-aware. In such cases, a DNS server can return DNSSEC data in a response. If the DO bit is not set, then client is not DNSSEC-aware. In such case, a DNS server cannot include any DNSSEC data in a response. Such clients can still be protected using DNSSEC.
- AD: The "Authenticated Data" bit is included in a DNS response sent from server to client. It indicates that the DNS response is authentic as it was validated using DNSSEC. If the AD response is not set, it indicates that either the DNSSEC validation was not performed or that the validation failed.
- CD: The "Checking Disabled" bit in a query indicates that a DNS response must be sent even if it will fail the validation. If the CD bit is not set, then the DNS response

will not be sent if it fails the validation. DNSSEC validation can occur only if the CD bit is clear (CD=0), that is, Checking Enabled.

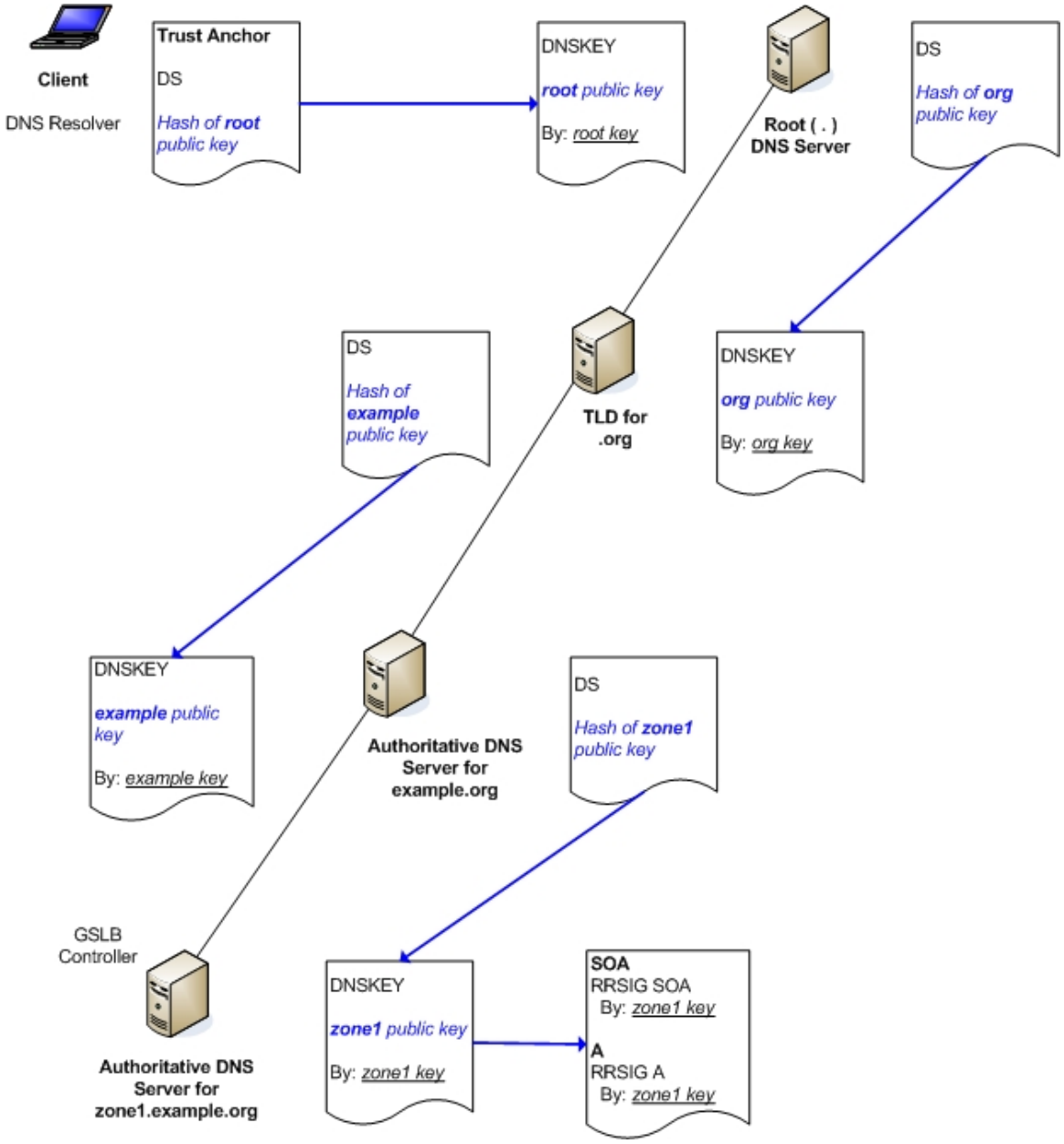
For details on the DNSSEC validation workflow, see the *Application Delivery Controller Guide*.

## Building the Chain of Trust

The DNSSEC Chain of Trust illustrates how the Chain of Trust is built in the DNSSEC infrastructure. A Chain of Trust is built like a series of links, where each node authenticates the one below it.

The presence of a Chain of Trust allows the client's DNS resolver to know that all of the DNS servers in the chain have vouched for one another, starting from the Root DNS Server and continuing down to the lowest-level DNS server.

Figure 7 : DNSSEC Chain of Trust




The DNSSEC Chain of Trust shows the Authoritative DNS Server for the *zone1.example.org* domain at the bottom left, and the Root DNS Server is located at the upper right.

Starting from the lower left, the Authoritative DNS Server for the *zone1.example.org* domain, has a DNS key record (DNSKEY). This DNSKEY record contains the public Zone

Signing Key (ZSK) for zone1. The ZSK is used to sign other record types, such as A records, for the zone. The DNSKEY record is signed by the Key Signing Key (KSK), which also belongs to this zone.

The Start of Authority (SOA) record indicates that this server is the Authoritative DNS Server for zone1. The A record provides the IP address for *zone1.example.org*.

The next level up in the DNS hierarchy corresponds to the next "label" in the *example.org* domain, and it has a record called the Delegation Signer (DS). The DS record contains a hash, or message digest, of the public Key Signing Key (KSK), which belongs to the Authoritative DNS Server for the node below, *zone1.example.org*.

The DNS resolver (or the Caching DNS Server) can compare the hash value for any of the nodes in the Chain of Trust, and the values should match. If the hash values in a DS record cannot be recreated from the DNSKEY record, packet that contains the key record may have been tampered with, cannot be trusted, and should be discarded.

However, if the hash value is correct, this indicates that the Chain of Trust is unbroken and that the DNSKEY record for the Authoritative DNS Server that is associated with the *zone1.example.org* domain is properly linked to the DS record above.

In turn, the DNSKEY record for the Authoritative DNS Server associated with the *example.org* domain is properly linked to the DS record above. This process of DNSKEY records being linked with the DS record of the node above continues all the way to the Root DNS Server.

The client's DNS resolver knows that the Root DNS Server is legitimate due to the presence of a "trust anchor". This trust anchor, which consists of information for the Root DNS Server, is included in the resolver software that is installed on the client. This minimizes the chance that a client could access a corrupt root DNS server.

Because of this anchor, the client knows that the Root DNS Server can be trusted, and the client can infer that the other nodes in the Chain of Trust can also be trusted. The hash values match all the way down the line, which is an indication that the Chain of Trust is intact, and that the client's DNS resolver can trust the Authoritative DNS Server for *zone1.example.org*. The Server is located at the bottom of the Chain of Trust in the DNS hierarchy.

# Dynamic Key Generation and Rollover

DNSSEC uses dynamic key generation and rollover that are provided by HSM, and HSM configuration is required.

The following topics are covered:



## Key Generation and Rollover Parameters

When HSM and DNSSEC are enabled, ACOS uses the following key generation and rollover settings for DNSSEC:

- **Key size** – Length of the keys in bits. You can specify 1024-4096 bits. The default length for ZSKs and KSKs is 2048 bits.
- **Lifetime** – Maximum amount of time a dynamically generated key remains valid.
- **Rollover time** – Amount of time to wait after a new key becomes active, before generating that key's replacement.

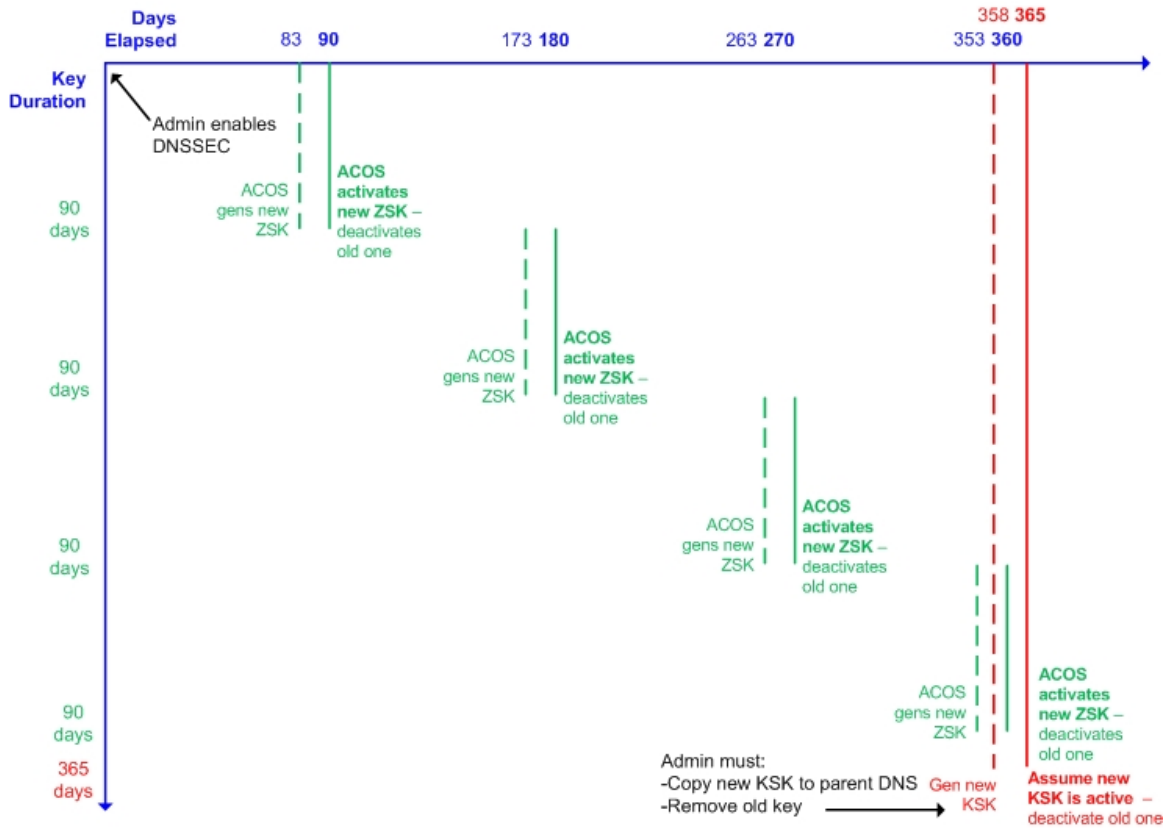
The range of values for the lifetime and rollover time is 1 to 2,147,483,647 seconds (about 68 years). The default lifetime and rollover time differ for ZSKs and KSKs:

- **ZSKs** – The default lifetime is 7,776,000 seconds (90 days), and the default rollover time is 7,171,200 seconds (83 days).
- **KSKs** – The default lifetime is 31,536,000 seconds (365 days), and the rollover time is 30,931,200 seconds (358 days).

## Key Rollover and Distribution Process

The features such as dynamic key generation and rollover are enabled by default when a DNSSEC template becomes active. No additional configuration is required. The DNSSEC - Default Rekey and Rollover shows the rekey and rollover schedule if the default rekey and rollover settings for ZSKs and KSKs are used.

Figure 8 : DNSSEC - Default Rekey and Rollover




When DNSSEC is enabled, HSM generates a KSK for the GSLB zone, generates a ZSK for the zone, and signs it with the KSK. The following text is an example of message that appears in the log.

## Key Regeneration Log Messages

ACOS generates messages such as the following when key regeneration occurs:

```
Log Buffer: 30000 Jul 31 2013 06:49:13 Notice  [DNS]:succeed to reload the
signature of zone "test.com"
Jul 31 2013 06:48:58 Notice  [CLI]: DNSSEC module:succeed to generate ZSK
test.com_zsk_2013-07-31-06-48-58 for zone test.com
Jul 31 2013 06:48:58 Notice  [CLI]: DNSSEC module:please transfer the DS
RR of zone test.com to the parent zone for the initial process.
Jul 31 2013 06:48:58 Notice  [CLI]: DNSSEC module:succeed to generate KSK
test.com_ksk_2013-07-31-06-48-57 for zone test.com
```

The first message, starting at the bottom, indicates a successful generation of a KSK for child zone test.com. The next message, which is second from the bottom, is a reminder to copy the DS resource record for the key to the authoritative DNS server for the parent zone.

The third message indicates a successful generation of the ZSK for child zone *test.com*. The final message at the top, indicates completion of the rekey process.

**CAUTION:** Although key generation and rollover are automatic, ACOS does not automatically send the DS record for the new KSK to the parent zone. This part of the process must be performed manually. If the default key generation and rollover settings are used, this process needs to be performed once a year.

## Importing/Exporting Key Files

The **import** command is used to import and export DNSSEC key files. For example, to import a file:

```
ACOS# import dnssec-dnskey zone-name
scp://exampleuser@examplehost.com/file
```

To export a file:

```
ACOS# export dnssec-dnskey zone-name
scp://exampleuser@examplehost.com/file
```

After enabling DNSSEC, wait about a minute for the key to be generated. You can use the **export dnssec-ds** command to copy the DS resource record for the zone to the DNS server that is authoritative for the parent zone.

For syntax information, see the *Command Line Interface Reference*.

## Emergency Key Rollover

The **dnssec key-rollover** command allows you to force an immediate key rollover, if necessary. For example, to force an immediate ZSK rollover in emergency mode:

```
ACOS(config)# dnssec key-rollover zone1 ZSK start
```

The **start** option initiates a rollover for the specified key type.

For KSK rollover, the **ds-ready-in-parent-zone** option indicates that the DS record for the new KSK has been exported to the parent zone. Use this option only after you have installed the DS record for the new KSK on the authoritative DNS server for the parent zone. For example:

```
ACOS(config)# dnssec key-rollover zone2 KSK ds-ready-in-parent-zone
```

## Changing Key Settings

Use the **zsk lifetime** and **ksk lifetime** commands to change the lifetime and rollover settings for ZSKs and KSKs, respectively.

Enter the commands at the configuration level for the DNSSEC template.

**NOTE:** For more information about the supported values, see Key Generation and Rollover Parameters.

# Hardware Security Module Support

Hardware Security Module (HSM) provides additional security, while simplifying key management.

The current release supports a software emulation version of HSM in ACOS. Keys are generated and stored on the ACOS device. This version can be useful for testing or for environments where the additional security of a hardware-based HSM is not required.

HSM is required for DNSSEC, and manual key generation of DNSSEC ZSKs or KSKs is not supported. For information about external HSM support, contact A10 Networks.

# DNSSEC

The following topics are covered:



# DNSSEC Configuration Example

The following topics are covered:



The following are the configuration modes from a device that is configured for DNSSEC.

## Configuring an HSM Template

The following commands configure an HSM template:

```
ACOS(config)# hsm template hsm1 softHSM
ACOS(config-template:hsm1)# password encrypted
/+mboU9rpJM8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
ACOS(config-template:hsm1)# exit
```

## Configuring a DNSSEC Template

The following commands configure a DNSSEC template:

```
ACOS(config)# dnssec template dt1
ACOS(config-dnssec)# zsk lifetime 2400 rollover-time 1900
ACOS(config-dnssec)# ksk lifetime 2500 rollover-time 2000
ACOS(config-dnssec)# signature-validity-period 11
ACOS(config-dnssec)# dnskey-ttl 5
ACOS(config-dnssec)# hsm hsm1
ACOS(config-dnssec)# exit
```

**NOTE:** ACOS checks the validity of DNSSEC signatures everyday. This ensures that if the signatures are due to expire in the next 1 or 2 days, they are duly resigned well on time.

## Configuring GSLB

The following commands configure GSLB.

```
ACOS(config)# gslb service-ip vip-1 1.0.0.1
ACOS(config-service-ip:vip-1)# health-check-protocol-disable
ACOS(config-service-ip:vip-1)# health-check-disable
ACOS(config-service-ip:vip-1)# port 80 tcp
ACOS(config-service-ip:vip-1-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-1-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-1-port:tcp)# exit
ACOS(config-service-ip:vip-1)# port 21 tcp
ACOS(config-service-ip:vip-1-port:tcp)# exit
ACOS(config-service-ip:vip-1)# exit
ACOS(config)# gslb service-ip vip-2 1.0.0.2
ACOS(config-service-ip:vip-2)# health-check-protocol-disable
ACOS(config-service-ip:vip-2)# health-check-disable
```

```
ACOS(config-service-ip:vip-2)# port 80 tcp
ACOS(config-service-ip:vip-2-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-2-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-2-port:tcp)# exit
ACOS(config-service-ip:vip-2)# port 21 tcp
ACOS(config-service-ip:vip-2-port:tcp)# exit
ACOS(config-service-ip:vip-2)# exit
ACOS(config)# gslb service-ip vip-3 1.0.0.3
ACOS(config-service-ip:vip-3)# health-check-protocol-disable
ACOS(config-service-ip:vip-3)# health-check-disable
ACOS(config-service-ip:vip-3)# port 80 tcp
ACOS(config-service-ip:vip-3-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-3-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-3-port:tcp)# exit
ACOS(config-service-ip:vip-3)# port 21 tcp
ACOS(config-service-ip:vip-3-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-3-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-3-port:tcp)# exit
ACOS(config-service-ip:vip-3)# exit
ACOS(config)# gslb service-ip ns 10.10.10.5
ACOS(config-service-ip:ns)# health-check-protocol-disable
ACOS(config-service-ip:ns)# health-check-disable
ACOS(config-service-ip:ns)# exit
ACOS(config)# gslb service-ip vip-4 1.0.0.4
ACOS(config-service-ip:vip-4)# health-check-protocol-disable
ACOS(config-service-ip:vip-4)# health-check-disable
ACOS(config-service-ip:vip-4)# port 80 tcp
ACOS(config-service-ip:vip-4-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-4-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-4-port:tcp)# exit
ACOS(config-service-ip:vip-4)# port 21 tcp
ACOS(config-service-ip:vip-4-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-4-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-4-port:tcp)# exit
ACOS(config-service-ip:vip-4)# exit
ACOS(config)# gslb service-ip vip-5 1.0.0.5
ACOS(config-service-ip:vip-5)# health-check-protocol-disable
ACOS(config-service-ip:vip-5)# health-check-disable
ACOS(config-service-ip:vip-5)# port 80 tcp
ACOS(config-service-ip:vip-5-port:tcp)# health-check-protocol-disable
```

```
ACOS(config-service-ip:vip-5-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-5-port:tcp)# exit
ACOS(config-service-ip:vip-5)# port 21 tcp
ACOS(config-service-ip:vip-5-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-5-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-5-port:tcp)# exit
ACOS(config-service-ip:vip-5)# exit
ACOS(config)# gslb service-ip vip-6 1.0.0.6
ACOS(config-service-ip:vip-6)# health-check-protocol-disable
ACOS(config-service-ip:vip-6)# health-check-disable
ACOS(config-service-ip:vip-6)# port 80 tcp
ACOS(config-service-ip:vip-6-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-6-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-6-port:tcp)# exit
ACOS(config-service-ip:vip-6)# port 21 tcp
ACOS(config-service-ip:vip-6-port:tcp)# health-check-protocol-disable
ACOS(config-service-ip:vip-6-port:tcp)# health-check-disable
ACOS(config-service-ip:vip-6-port:tcp)# exit
ACOS(config-service-ip:vip-6)# exit
ACOS(config)# gslb service-ip vip6-1 2001:111::1
ACOS(config-service-ip:vip6-1)# port 80 tcp
ACOS(config-service-ip:vip6-1-port:tcp)# exit
ACOS(config-service-ip:vip6-1)# port 21 tcp
ACOS(config-service-ip:vip6-1-port:tcp)# exit
ACOS(config-service-ip:vip6-1)# exit
ACOS(config)# gslb service-ip vip6-2 2001:111::2
ACOS(config-service-ip:vip6-2)# port 80 tcp
ACOS(config-service-ip:vip6-2-port:tcp)# exit
ACOS(config-service-ip:vip6-2)# port 21 tcp
ACOS(config-service-ip:vip6-2-port:tcp)# exit
ACOS(config-service-ip:vip6-2)# exit
ACOS(config)# gslb service-ip vip6-3 2001:111::3
ACOS(config-service-ip:vip6-3)# port 80 tcp
ACOS(config-service-ip:vip6-3-port:tcp)# exit
ACOS(config-service-ip:vip6-3)# port 21 tcp
ACOS(config-service-ip:vip6-3-port:tcp)# exit
ACOS(config-service-ip:vip6-3)# exit
ACOS(config)# gslb service-ip vip6-4 2001:111::4
ACOS(config-service-ip:vip6-4)# port 80 tcp
ACOS(config-service-ip:vip6-4-port:tcp)# exit
```

```
ACOS(config-service-ip:vip6-4)# port 21 tcp
ACOS(config-service-ip:vip6-4-port:tcp)# exit
ACOS(config-service-ip:vip6-4)# exit
ACOS(config)# gslb service-ip vip6-5 2001:111::5
ACOS(config-service-ip:vip6-5)# port 80 tcp
ACOS(config-service-ip:vip6-5-port:tcp)# exit
ACOS(config-service-ip:vip6-5)# port 21 tcp
ACOS(config-service-ip:vip6-5-port:tcp)# exit
ACOS(config-service-ip:vip6-5)# exit
ACOS(config)# gslb service-ip vip6-6 2001:111::6
ACOS(config-service-ip:vip6-6)# port 80 tcp
ACOS(config-service-ip:vip6-6-port:tcp)# exit
ACOS(config-service-ip:vip6-6)# port 21 tcp
ACOS(config-service-ip:vip6-6-port:tcp)# exit
ACOS(config-service-ip:vip6-6)# exit
ACOS(config)# gslb service-ip vip-187 1.1.1.187
ACOS(config-service-ip:vip-187)# health-check-protocol-disable
ACOS(config-service-ip:vip-187)# health-check-disable
ACOS(config-service-ip:vip-187)# exit
ACOS(config)# gslb site local
ACOS(config-gslb site:local)# bw-cost limit 100 threshold 10
ACOS(config-gslb site:local)# slb-dev self 127.0.0.1
ACOS(config-gslb site:local-slb dev:self)# vip-server vip1
ACOS(config-gslb site:local-slb dev:self)# vip-server vip2
ACOS(config-gslb site:local-slb dev:self)# vip-server vip3
ACOS(config-gslb site:local-slb dev:self)# exit
ACOS(config-gslb site:local)# ip-server ns
ACOS(config-gslb site:local)# ip-server vip-187
ACOS(config-gslb site:local)# ip-server vip-1
ACOS(config-gslb site:local)# ip-server vip-2
ACOS(config-gslb site:local)# ip-server vip-3
ACOS(config-gslb site:local)# exit
ACOS(config)# gslb site remote
ACOS(config-gslb site:remote)# weight 10
ACOS(config-gslb site:remote)# slb-dev site 192.168.217.1
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip6-4
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip6-5
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip6-6
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip-4
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip-5
```

```
ACOS(config-gslb site:remote-slb dev:site)# vip-server vip-6
ACOS(config-gslb site:remote-slb dev:site)# exit
ACOS(config-gslb site:remote)# exit
ACOS(config)#
```

## Configuring a GSLB Policy and Enable Server Mode

The **gslb policy** command configures a GSLB policy.

```
ACOS(config)# gslb policy gpol1
ACOS(config-policy:gpol1)# dns geoloc-alias
ACOS(config-policy:gpol1)# dns server authoritative ns ptr srv sec
```

The **dns server** command enables server mode, and also enables this ACOS device to be the authoritative DNS server for the GSLB zones that use this policy.

## Binding the DNSSEC Template to the Zone

Use the **template dnssec** command to bind the DNSSEC template to the zone:

```
ACOS(config)# gslb zone test.com
ACOS(config-zone:test.com)# policy gpol1
ACOS(config-zone:test.com)# template dnssec dt1
ACOS(config-zone:test.com)# service 0 www
ACOS(config-zone:test.com-service:www)# dns-a-record vip-2 static
ACOS(config-zone:test.com-service:www)# dns-a-record vip-1 static
ACOS(config-zone:test.com-service:www)# exit
ACOS(config-zone:test.com)# exit
ACOS(config)# gslb zone test1.com
ACOS(config-zone:test.com)# policy gpol1
ACOS(config-zone:test.com)# template dnssec dt1
ACOS(config-zone:test.com)# service 0 www
ACOS(config-zone:test.com-service:www)# dns-a-record vip-2 static
ACOS(config-zone:test.com-service:www)# dns-a-record vip-1 static
ACOS(config-zone:test.com-service:www)# exit
ACOS(config-zone:test.com)# exit
```

## Configuring DNSSEC Standalone

The ACOS device does not need to be a member of a GSLB controller group to run DNSSEC. GSLB is still required with standalone DNSSEC operation, but configuring a GSLB controller group is not required.

By default, support for standalone DNSSEC operation is optional and is disabled.

```
ACOS(config)# dnssec standalone
```

## Configuring the VIP for DNSSEC Requests

The following commands configure the virtual servers and DNS service ports:

```
ACOS(config)# slb virtual-server vs-1 10.105.1.111
ACOS(config-slb vserver)# port 53  udp
ACOS(config-slb vserver-vport)# name _1.1.1.1_UDP_53
ACOS(config-slb vserver-vport)# gslb-enable
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# port 53  dns-tcp
ACOS(config-slb vserver-vport)# name _1.1.1.1_DNS-TCP_53
ACOS(config-slb vserver-vport)# gslb-enable
ACOS(config-slb vserver-vport)# exit
ACOS(config-slb vserver)# exit
```

# Location-Based VIP Access

The following topics are covered:

# Overview of Location-based VIP Access

You can control access to a VIP that is based on the geo-location of the client. Depending on the location of the client, you also can configure ACOS to perform one of the following actions for traffic from a client:

- Drop the traffic
- Reset the connection
- Send the traffic to a specific service group (if configured using a black/white list)

ACOS determines a client's location by looking up the client's subnet in the geo-location database that is used by Global Server Load Balancing (GSLB).

**NOTE:** This feature requires you to load a geo-location database, but does not require any other configuration of GSLB. The ACOS system image includes the Internet Assigned Numbers Authority (IANA) database. By default, the IANA database is not loaded but you can easily load it. For more information, see Loading the IANA Geo-Location Database.

# Configuration Using a Class List

This section shows how to configure the geo-location-based VIP access by using a class list.

**NOTE:** In the current release, geo-location-based VIP access works only if the class list is imported as a file. The CLI does not support configuration of class-list entries for this application.

## Example

The following class list maps client geo-locations to limit IDs (LIDs), which specify the maximum number of concurrent connections allowed for clients in the geo-locations.

```
L US 1
L US.CA 2
```

```
L US.CA.SJ 3
```

The following commands import the class list to the ACOS device, configure a policy template, and bind the template to a virtual port. The connection limits specified in the policy template apply to clients that send requests to the virtual port.

**NOTE:** This example assumes the default geo-location database (iana) is loaded.

```
ACOS(config)# import class-list c-share tftp://192.168.32.162/
File name [/]? c-share
Importing ... Done.
ACOS(config)# slb template policy pclass
ACOS(config-policy)# class-list c-share
ACOS(config-policy-class-list:c-share)# lid 1
ACOS(config-policy-class-list:c-share-li...)# conn-limit 4
ACOS(config-policy-class-list:c-share-li...)# exit
ACOS(config-policy-class-list:c-share)# lid 2
ACOS(config-policy-class-list:c-share-li...)# conn-limit 2
ACOS(config-policy-class-list:c-share-li...)# exit
ACOS(config-policy-class-list:c-share)# lid 3
ACOS(config-policy-class-list:c-share-li...)# conn-limit 1
ACOS(config-policy-class-list:c-share-li...)# exit
ACOS(config-policy-class-list:c-share)# exit
ACOS(config-policy)# geo-location overlap
ACOS(config-policy)# exit
ACOS(config)# slb virtual-server vip1 10.1.1.155
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy pclass
ACOS(config-slb vserver-vport)# exit
```

The following command verifies the operation of the policy:

```
ACOS(config-policy)# show slb geo-location statistics

                          M = Matched or Level, ID = Group ID
                          Conn = Connection number, Last = Last Matched IP
                            v = Exact Match, x = Fail
```

```
Virtual Server: vip1/80, c-share
------------------------------------------------------------------------------
------
 max Depth: 3
   Success: 3
Geo-location                  M  ID Permit    Deny      Conn      Last

------------------------------------------------------------------------------
------
US.CA.SJ                      v  3  1         1          1
       77.1.1.107
------------------------------------------------------------------------------
------
     Total: 1
```

# Configuration Using a Black/White List

The following topics are covered:



## Details

To configure geo-location-based access control for a VIP:

1. Configure a black/white list.

   You can configure the list by using a text editor or enter the list into the GUI. If you configure the list by using a text editor, import the list to the ACOS device.

2. Configure an SLB policy (PBSLB) template.

In the template, specify the black/white list name, and the actions to perform for the group IDs in the list.

3. Verify that the geo-location database is loaded.

   For more information about loading the geo-location database, see Loading the IANA Geo-Location Database.

4. Apply the policy template to the virtual port for which you want to control access.

## Configuring the Black/White List

The following topics are covered:



### Methods

You can configure black/white lists in one of the following ways:

- **Remote** – Use a text editor and import the list to the ACOS device.
- **Local** – Enter the black/white list in a management GUI window.

With both methods, the syntax is the same. The black/white list must be a text file that contains entries (rows) in the following format:

```
L "geo-location" group-id #conn-limit
```

The various parameters in the syntax are described in the Black/White List Syntax Description.

Black/White List Syntax Description

| Parameter | Description |
|---|---|
| **L** | Indicates that the client's location will be determined by using information in the geo-location database. |

Black/White List Syntax Description

| Parameter | Description |
|---|---|
| *geo-location* | String in the geo-location database that is mapped to the client's IP address, for example, "US", "US.CA", or "US.CA.SanJose". |
| *group-id* | Number from 1 to 31 that identifies a group of clients (geo-locations) in the list. The default group ID is 0, which means no group is assigned. On the ACOS device, the group ID specifies the action to perform on client traffic. |
| *#conn-limit* | Maximum number of concurrent connections allowed from a client. The # is required only if you do not specify a group ID. The connection limit is optional. For simplicity, the examples in this section do not specify a connection limit. |

Below is a simple example of a Black/White list:

```
L "US"          1
L "US.CA"       2
L "JP"          3
```

## Using the GUI

The following topics are covered:

### Creating a Black-White List

To configure Black-White list by using the GUI:

1. Navigate to **ADC >> Black-white Lists**.
2. Click **Create** and complete the fields on the Create Black-White List page.

   Enter the list in the Definition field.

**NOTE:** For more details and information about any of the required fields on this page, see the latest version of the **GUI Online Help**.

3. Click **Create**.

## Configuring an SLB policy (PBSLB) Template

To configure an SLB policy template:

1. Navigate to **ADC >> Templates > L7**.
2. Click **Create** and select **Policy** from the drop-down list.
3. In the Name field, specify a template name.
4. Complete the other fields on the screen as desired.

**NOTE:** For more details and information about any of the required fields on this page, see the latest version of the **GUI Online Help**.

5. Click **OK**.

## Loading the IANA Geo-Location Database

To load the IANA geo-location database:

1. Navigate to **GSLB >> Geo-Location Files**.
2. Click **Import**.
3. Specify **iana** in the Name field.
4. Complete the **Host** and **Location** fields to specify the location of the file you are importing.
5. Leave the Template fields blank.
6. Click **Import**.

**NOTE:** You can also import a custom geo-location database.

**NOTE:** For more information, see the *Global Server Load Balancing Guide*.

## Applying the Policy Template to a Virtual Port

To apply the policy template to a new virtual port:

1. Navigate to **ADC >> SLB >> Virtual Servers**.
2. Click **Create**.
3. Specify the name and IP address of the virtual server.
4. In the Virtual Port section, click **Create**.
    a. Specify a protocol and port number.
    b. Expand the Templates section.
    c. In the Template Policy field, select the desired policy template.
    d. Click **Create**.
    e. Click **Update**.

### CLI Example

The following command imports black/white list "**geolist**" onto the ACOS device.

```
ACOS(config)# import bw-list geolist scp://192.168.1.2/root/geolist
```

The following commands configure a policy template named "**geoloc**" and add the black/white list to it. The template is configured to drop traffic from clients in the geo-location mapped to group 1 in the list.

```
ACOS(config)# slb template policy geoloc
ACOS(config-policy)# bw-list name geolist
ACOS(config-policy)# bw-list id 1 drop
ACOS(config-policy)# exit
```

The following commands apply the policy template to port 80 on virtual server "**vip1**":

```
ACOS(config)# slb virtual-server vip1
ACOS(config-slb vserver)# port 80 http
ACOS(config-slb vserver-vport)# template policy geoloc
```

To view SLB geo-location statistics, use the **`show slb geo-location`** command.

# Enabling Full-Domain Checking

The following topics are covered:



## Details

By default, when a client requests a connection, the ACOS device checks the connection count only for the specific geo-location level of the client. If the connection limit for that specific geo-location level is not reached, the client's connection is permitted. Similarly, the permit counter is increased only for that specific geo-location level.

Geo-location connection limit example shows an example set of geo-location connection limits and current connections.

Geo-location connection limit example

| Geo-location | Connection Limit | Current Connections |
|---|---|---|
| US | 100 | 100 |
| US.CA | 50 | 37 |
| US.CA.SanJose | 20 | 19 |

Using the default behavior, the connection request from the client at US.CA.SanJose is allowed even though CA has reached its connection limit. Similarly, a connection request from a client at US.CA is allowed. However, a connection request from a client whose location match is simply "**US**" is denied.

After these three clients are permitted or denied, the connection permit and deny counters are increased in the following way:

- **US** – Deny counter is increased by 1.
- **US.CA** – Permit counter is increased by 1.
- **US.CA.SanJose** – Permit counter is increased by 1.

When full-domain checking is enabled, the ACOS device checks the current connection count not only for the client's specific geo-location, but for all geo-locations higher up in the domain tree.

Based on full-domain checking, all three connection requests from the clients in the example above are denied. This is because the US domain has reached its connection limit. Similarly, the counters for each domain are updated as follows:

- **US** – Deny counter is incremented by 1.
- **US.CA** – Deny counter is incremented by 1.

## Using the GUI to Configure Full-Domain Checking

This is configurable on the configuration page for the policy template:

1. Navigate to **ADC >> Templates >> L7**.
2. Click **Create** and select **Policy** from the drop-down list.
3. Specify a name for the policy template.
4. Expand the Geo Location pane.
5. Select **Full Domain Tree**.
6. Click **OK**.

## Using the CLI to Configure Full-Domain Checking

To enable full-domain checking for geo-location-based connection limiting, enter the **`geo-location full-domain-tree`** command at the configuration level for the PBSLB template:

```
ACOS(config)# slb template policy example_policy_template
ACOS(config-policy)# geo-location full-domain-tree
```

**NOTE:** You must enable or disable this option before you enable GSLB. Changing the state of this option while GSLB is running can cause the related statistics counters to be incorrect.

# Enabling PBSLB Statistics Counter Sharing

The following topics are covered:



## Details

You can enable sharing of statistics counters for all virtual servers and virtual ports that use a PBSLB template. This option causes the following counters to be shared by the virtual servers and virtual ports that use the template:

- Permit
- Deny
- Connection number
- Connection limit

## Using the GUI to Enable PBSLB Statistics Counter Sharing

This is configurable on the configuration page for the policy template:

1. Navigate to **ADC >> Templates >> L7**.
2. Click **Create** and select **Policy** from the drop-down list.
3. Specify a name for the policy template.
4. Expand the Geo Location pane.

5. Select **Share**.
6. Click **OK**.

## Using the CLI to Enable PBSLB Statistics Counter Sharing

To enable the share option, enter the **geo-location share** command at the configuration level for the PBSLB policy template:

```
ACOS(config)# slb template policy example_policy_template
ACOS(config-policy)# geo-location share
```

**NOTE:** You must enable or disable this option before you enable GSLB. Changing the state of this option while GSLB is running can cause the related statistics counters to be incorrect.

## Contact Us