CHAPTER 51

Configuring TCP State Bypass

This chapter describes how to configure TCP state bypass, which lets outbound and inbound flows go through separate ASAs. This chapter includes the following sections:

- Information About TCP State Bypass, page 51-1
- Licensing Requirements for TCP State Bypass, page 51-2
- Guidelines and Limitations, page 51-2
- Default Settings, page 51-3
- Configuring TCP State Bypass, page 51-3
- Monitoring TCP State Bypass, page 51-4
- Configuration Examples for TCP State Bypass, page 51-4
- Feature History for TCP State Bypass, page 51-5

Information About TCP State Bypass

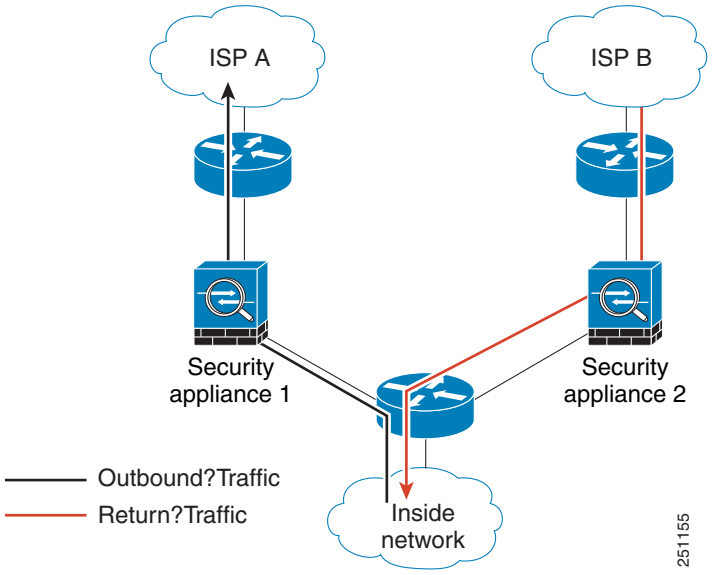
By default, all traffic that goes through the ASA is inspected using the Adaptive Security Algorithm and is either allowed through or dropped based on the security policy. The ASA maximizes the firewall performance by checking the state of each packet (is this a new connection or an established connection?) and assigning it to either the session management path (a new connection SYN packet), the fast path (an established connection), or the control plane path (advanced inspection). See the "Stateful Inspection Overview" section on page 1-13 for more detailed information about the stateful firewall.

TCP packets that match existing connections in the fast path can pass through the ASA without rechecking every aspect of the security policy. This feature maximizes performance. However, the method of establishing the session in the fast path using the SYN packet, and the checks that occur in the fast path (such as TCP sequence number), can stand in the way of asymmetrical routing solutions: both the outbound and inbound flow of a connection must pass through the same ASA.

For example, a new connection goes to ASA 1. The SYN packet goes through the session management path, and an entry for the connection is added to the fast path table. If subsequent packets of this connection go through ASA 1, then the packets will match the entry in the fast path, and are passed through. But if subsequent packets go to ASA 2, where there was not a SYN packet that went through

the session management path, then there is no entry in the fast path for the connection, and the packets are dropped. Figure 51-1 shows an asymmetric routing example where the outbound traffic goes through a different ASA than the inbound traffic:

Figure 51-1 Asymmetric Routing



If you have asymmetric routing configured on upstream routers, and traffic alternates between two ASAs, then you can configure TCP state bypass for specific traffic. TCP state bypass alters the way sessions are established in the fast path and disables the fast path checks. This feature treats TCP traffic much as it treats a UDP connection: when a non-SYN packet matching the specified networks enters the ASA, and there is not an fast path entry, then the packet goes through the session management path to establish the connection in the fast path. Once in the fast path, the traffic bypasses the fast path checks.

Licensing Requirements for TCP State Bypass

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent mode.

Failover Guidelines

Failover is supported.

Unsupported Features

The following features are not supported when you use TCP state bypass:

- Application inspection—Application inspection requires both inbound and outbound traffic to go through the same ASA, so application inspection is not supported with TCP state bypass.
- AAA authenticated sessions—When a user authenticates with one ASA, traffic returning via the other ASA will be denied because the user did not authenticate with that ASA.
- TCP Intercept, maximum embryonic connection limit, TCP sequence number randomization—The ASA does not keep track of the state of the connection, so these features are not applied.
- TCP normalization—The TCP normalizer is disabled.
- SSM and SSC functionality—You cannot use TCP state bypass and any application running on an SSM or SSC, such as IPS or CSC.

NAT Guidelines

Because the translation session is established separately for each ASA, be sure to configure static NAT on both ASAs for TCP state bypass traffic; if you use dynamic NAT, the address chosen for the session on ASA 1 will differ from the address chosen for the session on ASA 2.

Default Settings

TCP state bypass is disabled by default.

Configuring TCP State Bypass

This section describes how to configure TCP state bypass.

	Command	Purpose
Step 1	**class-map** *name* **Example:** `hostname(config)# class-map bypass_traffic`	Creates a class map to identify the traffic for which you want to disable stateful firewall inspection.
Step 2	**match** *parameter* **Example:** `hostname(config-cmap)# match access-list bypass`	Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 9-13 for more information.
Step 3	**policy-map** *name* **Example:** `hostname(config)# policy-map tcp_bypass_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic.

	Command	Purpose
Step 4	**class** *name* **Example:** hostname(config-pmap)# class bypass_traffic	Identifies the class map you created in Step 1
Step 5	**set connection advanced-options tcp-state-bypass** **Example:** hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass	Enables TCP state bypass.
Step 6	**service-policy** *policymap_name* {**global** | **interface** *interface_name*} **Example:** hostname(config)# service-policy tcp_bypass_policy outside	Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Monitoring TCP State Bypass

To monitor TCP state bypass, perform one of the following tasks:

Command	Purpose
show conn	If you use the **show conn** command, the display for connections that use TCP state bypass includes the flag "b."

Configuration Examples for TCP State Bypass

The following is a sample configuration for TCP state bypass:

```
hostname(config)# access-list tcp_bypass extended permit tcp 10.1.1.0 255.255.255.224 any

hostname(config)# class-map tcp_bypass
hostname(config-cmap)# description "TCP traffic that bypasses stateful firewall"
hostname(config-cmap)# match access-list tcp_bypass

hostname(config-cmap)# policy-map tcp_bypass_policy
hostname(config-pmap)# class tcp_bypass
hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass

hostname(config-pmap-c)# service-policy tcp_bypass_policy outside

hostname(config-pmap-c)# static (inside,outside) 209.165.200.224 10.1.1.0 netmask
255.255.255.224
```

Feature History for TCP State Bypass

Table 51-1 lists the release history for this feature.

Table 51-1 Feature History for TCP State Bypass

Feature Name	Releases	Feature Information
TCP state bypass	8.2(1)	This feature was introduced. The following command was introduced: **set connection advanced-options tcp-state-bypass.**