Configure ZBFW from SD-WAN CLI Template

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

Configure

- Network Diagram
- Configuration
  - Control Plane
  - Data Plane

Verify

Introduction

This document describes how to Configure Zone-based Firewall (ZBFW) Policy Using a CLI Add-On Feature Template from Cisco Catalyst SD-WAN Manager.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Catalyst Software-Defined Wide Area Network (SD-WAN)
- Zone-Based Firewall (ZBFW) basic operation

Components Used

- Cisco Catalyst SD-WAN Manager 20.9.3.2
- Cisco IOS® XE Catalyst SD-WAN Edges 17.6.5a

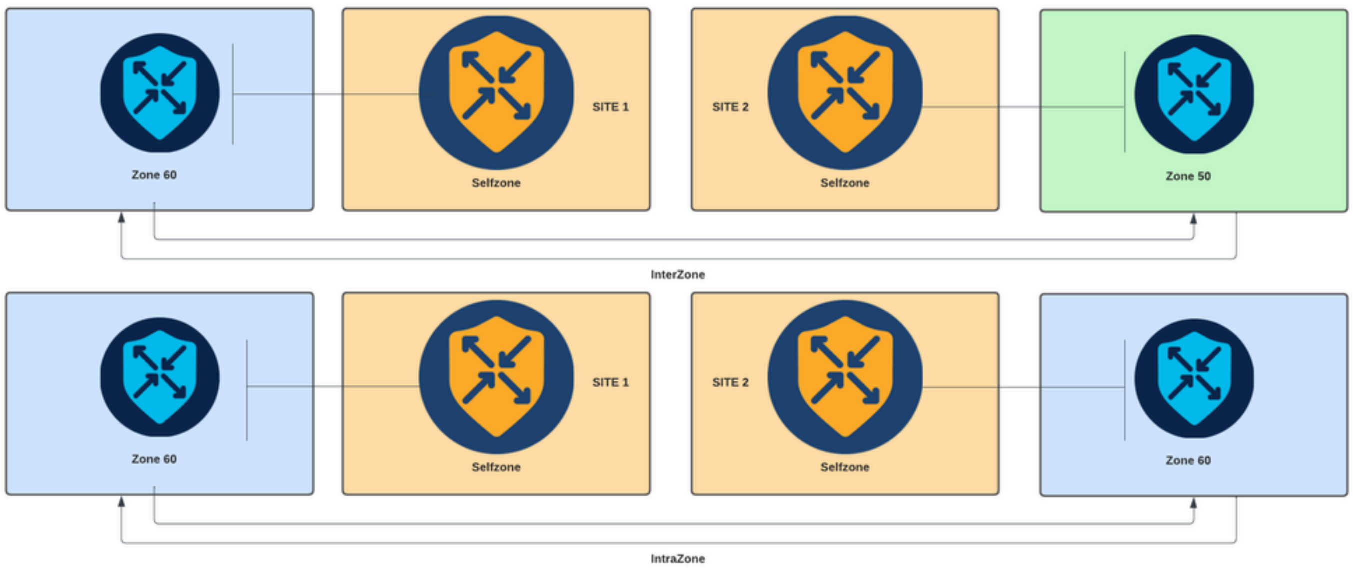
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

A firewall policy is a type of localized security policy that allows stateful inspection of TCP, UDP, and ICMP data traffic flows. It uses the concept of zones; therefore, traffic flows that originate in a given zone are allowed to proceed to another zone based on the policy between the two zones.

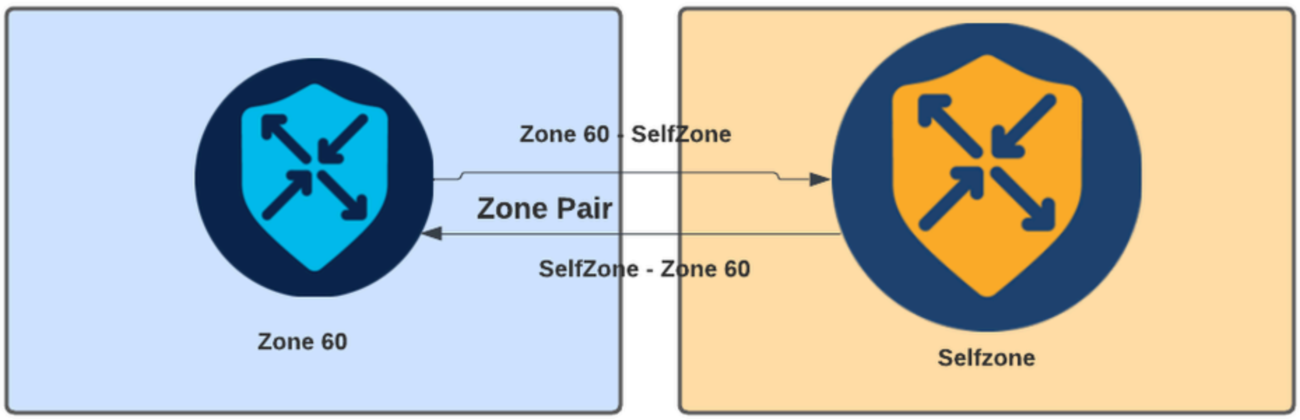
A zone is a group of one or more VPNs. The type of zones that exists on ZBFW are:

- **Source zone**: a group of VPNs that originates the data traffic flows. A VPN can be part of only one zone.
- **Destination zone**: a group of VPNs that terminates the data traffic flows. A VPN can be part of only one zone.
- **Interzone**: it is called interzone when the traffic flow between different zones (By default the communication is denied).
- **Intrazone**: it is called intrazone when the traffic flow through the same zone (By default the communication is permitted).
- **Selfzone**: it is used for controlling traffic that is sourced from or directed to the router itself (Default zone created and preconfigured by system, by default the communication is permitted).



Zone-Based Firewall Diagram

Another concept used in ZBFW is the zone pair, which is a container that associates a source zone with a destination zone. Zone pairs apply a firewall policy to the traffic that flows between the two zones.



Zone-Pair Example

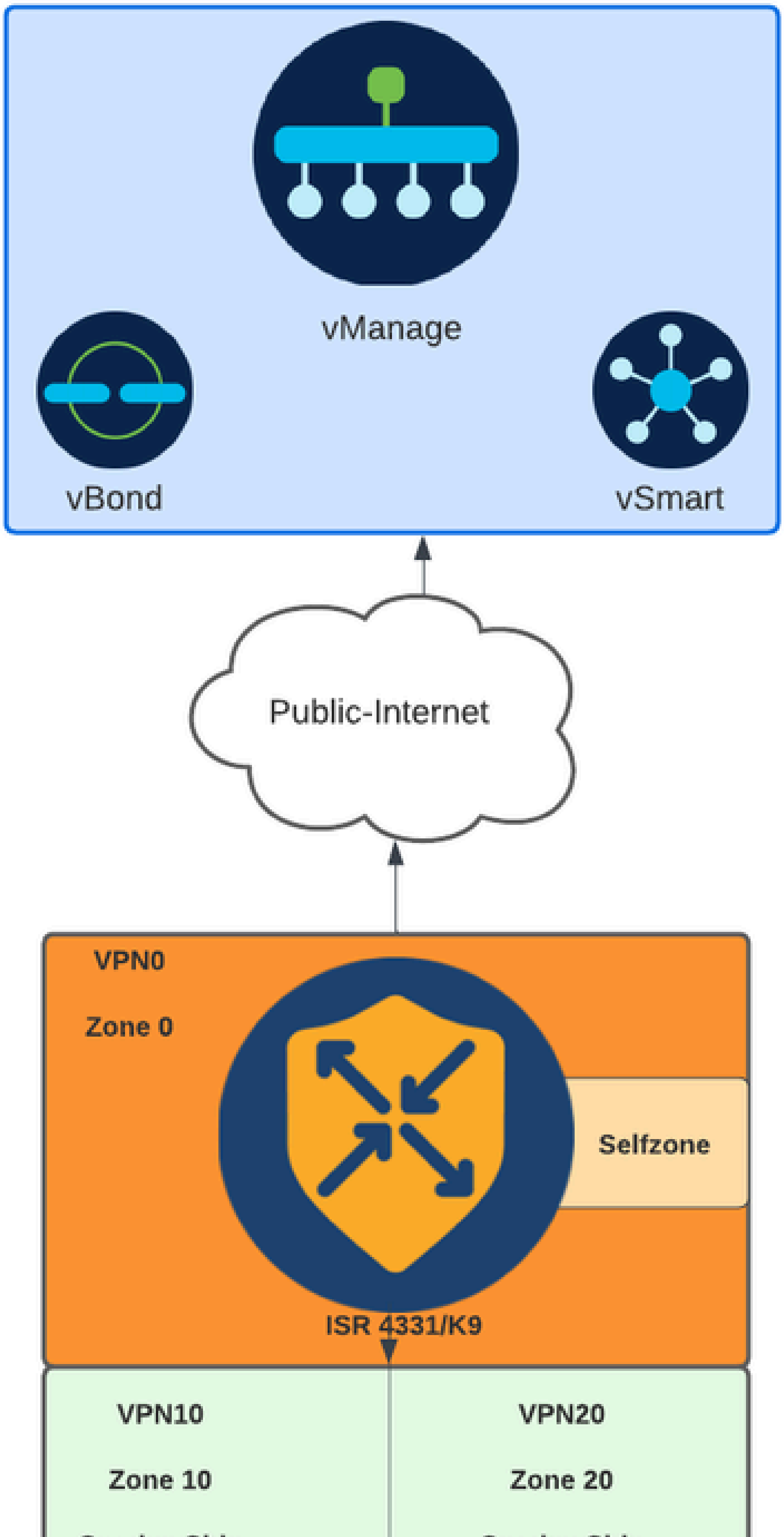
After the zone pair is defined, the actions that apply to the flows are:

- **Drop:** simply discards match flow.
- **Pass:** permits packet flow without stateful inspection, similar to the permit action in access-

lists. Whether a pass action is set in a flow, a return pass for that flow is needed.
- **Inspec**t: allows for stateful inspection of traffic that flows from source to destination zone, and automatically permits traffic flows to return.

Configure

Network Diagram


vManage
vBond
vSmart
Public-Internet
VPN0
Zone 0
Selfzone
ISR 4331/K9
VPN10
Zone 10
VPN20
Zone 20

: Whether WAN interface is configured via DHCP, it is necessary to create a rule to allow self zone (interface) to reach next-hop IP address in case reload device and router needs to get a new IP address.

Control Plane

1. Create the **inspect parameter map**:

```
 parameter-map type inspect-global
 multi-tenancy
 vpn zone security
  alert on
  log dropped-packets
  max-incomplete tcp timeout
```

The max-incomplete tcp <timeout> configuration command is used to specify the maximum number of incomplete connections before the TCP session is drop.

The **multi-tenancy** configuration command is a global parameter required in the ZBFW configuration. When ZBFW is configured via SD-WAN Manager GUI, the line is added by default. When ZBFW is configured via Command Line Interface (CLI), this line needs to be added.

2. Create a **WAN zone**:

```
zone security wan
vpn 0
```

Note: Self zone is created by default, it is not necessary to configure it.

3. Configure the **object group** for the source and destination addresses:

```
object-group network CONTROLLERS
 host 172.18.121.103
 host 172.18.121.106
 host 192.168.20.152
 host 192.168.22.203
object-group network WAN_IPs
 host 10.122.163.207
```

4. Create the **IP access-list**:

```
ip access-list extended self-to-wan-acl
 10 permit tcp object-group WAN_IPs object-group CONTROLLERS
 20 permit udp object-group WAN_IPs object-group CONTROLLERS
 30 permit ip object-group WAN_IPs object-group CONTROLLERS
ip access-list extended wan-to-self-acl
 10 permit tcp object-group CONTROLLERS object-group WAN_IPs
 20 permit udp object-group CONTROLLERS object-group WAN_IPs
 30 permit ip object-group CONTROLLERS object-group WAN_IPs
```

5. Create the **class map**:

```
class-map type inspect match-all self-to-wan-cm
 match access-group name self-to-wan-acl
class-map type inspect match-all wan-to-self-cm
 match access-group name wan-to-self-acl
```

6. Create the **policy map** to add to the **zone pair**:

```
policy-map type inspect wan-to-self-pm
 class type inspect wan-to-self-cm
  inspect
 class class-default
policy-map type inspect self-to-wan-pm
 class type inspect self-to-wan-cm
  inspect
 class class-default
```

7. Create the **zone pair** and link the **policy map** to it:

```
zone-pair security self-to-wan source self destination wan
 service-policy type inspect self-to-wan-pm
zone-pair security wan-to-self source wan destination self
 service-policy type inspect wan-to-self-pm
```

Once the control plane flows are allowed, data plane configuration can be applied.

To validate **control-connections**, use the EXEC command:

<#root>

```
Device#

show sdwan control connections
```

Whether ZBFW for self-zone and wan-zone is not correctly configure, the devices lose the control connections and get a console error similar the next:

<#root>

```
*Oct 30 19:44:17.731: %IOSXE-6-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:000 TS:00000004865486441431 %FW-6-
```

Data Plane

1. Create **security zone** for each Virtual Routing and Forwarding (VRF) needed:

```
zone security user
vpn 10
zone security server
vpn 20
```

3. Configure the **object group** for the source and destination addresses:

```
 object-group network USER
 host 10.10.10.1
 host 10.10.10.2
 host 10.10.10.3
object-group network SERVER
 host 10.20.20.1
 host 10.20.20.2
```

4. Create the **IP access-list**:

```
ip access-list extended user-to-server-acl
 10 permit tcp object-group USER object-group SERVER
 20 permit udp object-group USER object-group SERVER
 30 permit ip object-group USER object-group SERVER
ip access-list extended server-to-user-acl
 10 permit tcp object-group SERVER object-group USER
 20 permit udp object-group SERVER object-group USER
 30 permit ip object-group SERVER object-group USER
```

5. Create the **class map**:

```
class-map type inspect match-all user-to-server-cm
 match access-group name user-to-server-acl
class-map type inspect match-all server-to-wan-cm
 match access-group name server-to-user-acl
```

6. Create the **policy map** to add to the **zone pair**:

```
policy-map type inspect user-to-server-pm
```

```
 class type inspect user-to-server-cm
  inspect
 class class-default
policy-map type inspect server-to-user-pm
 class type inspect server-to-user-cm
  inspect
 class class-default
```

7. Create the **zone pair** and link the **policy map** to it:

```
zone-pair security user-to-server source user destination server
 service-policy type inspect user-to-server-pm
zone-pair security server-to-user source server destination user
 service-policy type inspect server-to-user-pm
```

Note: For more information about using CLI templates, see CLI Add-On Feature Templates and CLI Templates.

Verify

To validate the configured **inspect class-map**, use the EXEC command:

<#root>

```
Device#

show class-map type inspect
```

To validate the configured **inspect policy-map**, use the EXEC command:

<#root>

```
Device#

show policy-map type inspect
```

To validate the configured **zone-pair**, use the EXEC command:

<#root>

```
Device#

show zone-pair security
```

To validate the configured **access-list**, use the EXEC command:

<#root>

```
Device#

show ip access-list
```

To validate the configured **object-group**, use the EXEC command:

<#root>

```
Device#

show object-group
```

To display the **ZBFW session status**, use the EXEC command:

<#root>

```
Device#

show sdwan zonebfwdp sessions

 SRC DST TOTAL TOTAL UTD
SESSION SRC DST SRC DST VPN VPN NAT INTERNAL INITIATOR RESPONDER APPLICATION POLICY
ID STATE SRC IP DST IP PORT PORT PROTOCOL VRF VRF ID ID ZP NAME CLASSMAP NAME FLAGS FLAGS BYTES BYTES T
--------------------------------------------------------------------------------------------------------------

8 open 172.18.121.106 10.122.163.207 48960 32168 PROTO_L4_UDP 0 0 0 65534 wan-to-self wan-to-self-cm - 0


5 open 10.122.163.207 172.18.121.106 32168 32644 PROTO_L4_UDP 0 0 65534 0 self-to-wan self-to-wan-cm - 0

7 open 10.122.163.207 172.18.121.103 32168 32168 PROTO_L4_UDP 0 0 65534 0 self-to-wan self-to-wan-cm - 0


6 open 172.18.121.106 10.122.163.207 60896 32168 PROTO_L4_UDP 0 0 0 65534 wan-to-self wan-to-self-cm - 0
9 open 10.122.163.207 172.18.121.106 32168 34178 PROTO_L4_UDP 0 0 65534 0 self-to-wan self-to-wan-cm - 0
```

To display the **zone-pair statistics**, use the EXEC command:

<#root>

```
Device#

show sdwan zbfw zonepair-statistics

zbfw zonepair-statistics user-to-server
src-zone-name user
dst-zone-name server
policy-name user-to-server-pm
fw-traffic-class-entry user-to-server-cm
zonepair-name user-to-server

class-action Inspect
```

```
pkts-counter 0
bytes-counter 0
attempted-conn 0
```

```
current-active-conn 0
```

```
max-active-conn 0
current-halfopen-conn 0
max-halfopen-conn 0
current-terminating-conn 0
max-terminating-conn 0
```

```
time-since-last-session-create 0
```

```
<snipped>
```

To display the **ZBFW drop statistics**, use the EXEC command:

<#root>

```
Device#
```

```
show sdwan zbfw drop-statistics
```

```
zbfw drop-statistics catch-all                          0
zbfw drop-statistics l4-max-halfsession                 0
zbfw drop-statistics l4-session-limit                   0
zbfw drop-statistics l4-scb-close                       0
<snipped>
```

```
zbfw drop-statistics insp-policy-not-present           0
```

```
zbfw drop-statistics insp-sess-miss-policy-not-present 0
```

```
zbfw drop-statistics insp-classification-fail          0
zbfw drop-statistics insp-class-action-drop            0
zbfw drop-statistics insp-policy-misconfigure          0
<snipped>
zbfw drop-statistics l4-icmp-err-policy-not-present    0
<snipped>
```

```
zbfw drop-statistics invalid-zone                      0
```

```
zbfw drop-statistics ha-ar-standby                      0
zbfw drop-statistics no-forwarding-zone                 0
<snipped>
```

```
zbfw drop-statistics no-zone-pair-present            105 <<< If no zone-pair configured
```

To display the **QuantumFlow Processor (QFP) drop statistics**, use the EXEC command:

<#root>

```
Device#

show platform hardware qfp active statistic drop

Last clearing of QFP drops statistics: never
-------------------------------------------------------------------------
Global Drop Stats                         Packets                  Octets
-------------------------------------------------------------------------
<snipped>
BFDoffload                                    194                   14388
<snipped>
FirewallBackpressure                            0                       0

FirewallInvalidZone                             0                       0


FirewallL4                                      1                      74
FirewallL4Insp                                372                   40957
FirewallL7                                      0                       0

FirewallNoForwardingZone                        0                       0



FirewallNoNewSession                            0                       0


FirewallNonsession                              0                       0
FirewallNotFromInit                             0                       0
FirewallNotInitiator                        11898                  885244

FirewallPolicy                                  0                       0
```

To display the **QFP firewall drops**, use the EXEC command:

<#root>

```
Device#

show platform hardware qfp active feature firewall drop all

-------------------------------------------------------------------------------
Drop Reason                                                            Packets
-------------------------------------------------------------------------------
TCP out of window                                                        0
TCP window overflow                                                      0
<snipped>
TCP - Half-open session limit exceed                                     0
Too many packet per flow                                                 0
<snipped>
ICMP ERR PKT:no IP or ICMP                                               0
ICMP ERR Pkt:exceed burst lmt                                            0
ICMP Unreach pkt exceeds lmt                                             0
ICMP Error Pkt invalid sequence                                          0
ICMP Error Pkt invalid ACK                                               0
ICMP Error Pkt too short                                                 0
Exceed session limit                                                     0
Packet rcvd in SCB close state                                           0
Pkt rcvd after CX req teardown                                           0
CXSC not running                                                         0
```

```
Zone-pair without policy                              0 <<< Existing zone-pair, but not


Same zone without Policy                              0 <<< Zone without policy configu

<snipped>

No Zone-pair found                                  105 <<< If no zone-pair configured
```