Configure Zone-Based Firewall (ZBFW) co-located with Cisco Unified Border Element (CUBE) Enterprise

Contents

Introduction

This document describes how to configure Zone-Based Firewall (ZBFW) co-located with Cisco Unified Border Element (CUBE) Enterprise.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

- Cisco router running Cisco IOS® XE 17.10.1a

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

- CUBE Enterprise and ZBFW co-location was not supported on Cisco IOS XE until 16.7.1+

- CUBE Enterprise only supports CUBE + ZBFW RTP-RTP media flows. See: CSCwe66293

- This document is not applicable to CUBE Media Proxy, CUBE Service Provider, MGCP or SCCP Gateways, Cisco SRST or ESRST Gateways, H323 Gateways, or other Analog/TDM Voice Gateways.

- For TDM/Analog Voice Gateways and ZBFW see the following document: https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/213550-troubleshoot-one-way-audio-problems-in-f.html

Network Diagram

The sample configuration will illustrate two logical network segmentations named INSIDE and OUTSIDE.

INSIDE contains a single IP network and OUTSIDE contains two IP networks.

Layer 3 Network Topology

```
Endpoint_A - Network A - Gig1 - CUBE - Gig3 - Network B - CUCM
                                          \_ Network C - Endpoint_B
```

Layer 7 Call Flow

```
Call Direction ======================================>
Endpoint_A > SIP > CUBE > SIP > CUCM > SIP > Endpoint_B
```

Layer 7 Media Flow

```
Endpoint_A <> RTP <> CUBE <> RTP <> Endpoint_B
```

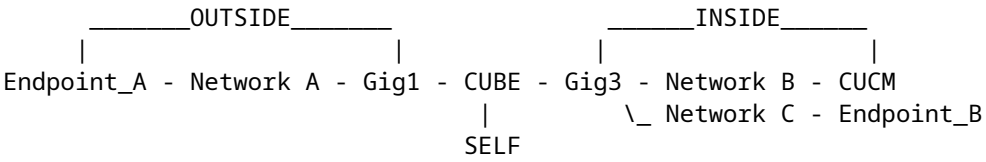
ZBFW Crash Course Concepts

- When configuring ZBFW you configure a security zone name which is then defined on an interface. After this all traffic to/from that interface is associated with that zone name.
  - Traffic to/from the same zone is always allowed.
  - Traffic to/from different zones is dropped unless allowed by administrator configuration.
- To define allowed traffic flows you must create a zone mapping via a unidirectional zone-pair configuration which defines the source and destination zone names.
  - This zone-pair mapping then ties into a service-policy used to provide granular control over the inspected, allowed, and disallowed traffic types.

- CUBE Enterprise operates in the special SELF zone. The SELF zone includes other traffic to/from the router such as ICMP, SSH, NTP, DNS, etc.
    - Hardware PVDM for use with CUBE LTI do not exist in the self zone and must be mapped to an administratively configured zone.
- ZBFW does not automatically allow return traffic so an administrator must configure zone pairs to define return traffic.

With the following 3 bullets in mind the following zones can be added overlayed on our L3 Network Topology where:

- Network A, Gig1 are the OUTSIDE zone
- Network B, Network C, and Gig3, are INSIDE zone
- CUBE is part of the SELF zone



Next we can logically create the four unidirectional zone-pair mappings we need for traffic flows through CUBE+ZBFW:

Source	Destination	Usage
OUTSIDE	SELF	Inbound SIP and RTP Media from Endpoint A
SELF	INSIDE	Outbound SIP and RTP Media from CUBE to CUCM and Endpoint B.
INSIDE	SELF	Inbound SIP and RTP media from CUCM and Endpoint B.
SELF	OUTSIDE	Outbound SIP and RTP media from CUBE to Endpoint A.

With these concepts in mind we can start configuring ZBFW on the Cisco IOS XE router acting as CUBE.

Configurations

Define Security Zones

Recall we need to configure two security zones: INSIDE and OUTSIDE. Self does not need to be defined as it is default.

```
!
zone security INSIDE
zone security OUTSIDE
!
```

Create access-list, class-map, and policy-map for trusted traffic

In order to control what traffic we must configure methods for the Router to match and permit.

To do this we will create an extended access-list, class-map and policy map that inspect our traffic.

For simplicity we will create a policy for each zone that maps both inbound and outbound traffic.

Note that configurations such as **match protocol sip** and **match protocol sip-tls** may be used but for illustrative purposes the IP/Ports have been configured

OUTSIDE Extended Access List, Class Map, Policy Map

<#root>

```
! Define Access List with ACLs for OUTSIDE interface


ip access-list extended TRUSTED-ACL-OUT
 10 remark Match SIP TCP/UDP 5060 and TCP TLS 5061
 11 permit tcp 192.168.1.0 0.0.0.255 any range 5060 5061
 12 permit tcp any 192.168.1.0 0.0.0.255 range 5060 5061
 13 permit udp 192.168.1.0 0.0.0.255 any eq 5060
 14 permit udp any 192.168.1.0 0.0.0.255 eq 5060
 !
 20 remark Match RTP Port Range, IOS-XE and Remote Endpoints
 21 permit udp 192.168.1.0 0.0.0.255 any range 8000 48198
 22 permit udp any 192.168.1.0 0.0.0.255 range 8000 48198
!
```

```
! Tie ACL with Class Map


class-map type inspect match-any TRUSTED-CLASS-OUT
 match access-group name TRUSTED-ACL-OUT
!
```

```
! Tie Class Map with Policy and inspect


policy-map type inspect TRUSTED-POLICY-OUT
 class type inspect TRUSTED-CLASS-OUT
  inspect
 class class-default
  drop log
!
```

INSIDE Extended Access List, Class Map, Policy Map

```
!
ip access-list extended TRUSTED-ACL-IN
 1 remark SSH, NTP, DNS
 2 permit tcp any any eq 22
 3 permit udp any any eq 123
 4 permit udp any any eq 53
 !
 10 remark Match SIP TCP/UDP 5060 and TCP TLS 5061
 11 permit tcp 192.168.2.0 0.0.0.255 any range 5060 5061
 12 permit tcp any 192.168.2.0 0.0.0.255 range 5060 5061
 13 permit udp 192.168.2.0 0.0.0.255 any eq 5060
 14 permit udp any 192.168.2.0 0.0.0.255 eq 5060
 !
 20 remark Match RTP Port Range, IOS-XE and Remote Endpoints
 21 permit udp 192.168.2.0 0.0.0.255 any range 8000 48198
 22 permit udp any 192.168.2.0 0.0.0.255 range 8000 48198
 23 permit udp 192.168.3.0 0.0.0.31 any range 8000 48198
 24 permit udp any 192.168.3.0 0.0.0.31 range 8000 48198
!
class-map type inspect match-any TRUSTED-CLASS-IN
 match access-group name TRUSTED-ACL-IN
!
policy-map type inspect TRUSTED-POLICY-IN
 class type inspect TRUSTED-CLASS-IN
  inspect
 class class-default
  drop log
!
```

Create Zone-Pair Mappings

Next we must create the four zone-pair mappings discussed earlier in the table.

These zone-pairs will reference a service policy which the policy-map we created earlier.

<#root>

```
! INSIDE <> SELF


zone-pair security IN-SELF source INSIDE destination self
 service-policy type inspect TRUSTED-POLICY-IN
zone-pair security SELF-IN source self destination INSIDE
 service-policy type inspect TRUSTED-POLICY-IN
!
```

```
! OUTSIDE <> SELF


zone-pair security OUT-SELF source OUTSIDE destination self
 service-policy type inspect TRUSTED-POLICY-OUT
zone-pair security SELF-OUT source self destination OUTSIDE
 service-policy type inspect TRUSTED-POLICY-OUT
!
```

Assign Zones to Interfaces

<#root>

```
! Assign Zones to interfaces


int gig1
 zone-member security INSIDE
!
int gig3
 zone-member security OUTSIDE
!
```

Verify

Sample Packet Flow - Call

At this point a call from Endpoint B to CUBE destined for CUCM will invoke the following sequence:

1. Inbound TCP SIP Packet to CUBE on 5060 will ingress GIG 1 and be mapped to OUTSIDE source zone
2. CUBE operates in SELF zone so the OUTSIDE to SELF zone-pair will be used (**OUT-SELF**)
3. The service-policy/policy-map **TRUSTED-POLICY-OUT** will be used to inspect traffic based on **TRUSTED-CLASS-OUT** class-map and **TRUSTED-ACL-OUT** access-list
4. CUBE will then use local call routing logic to determine where to send the call and what egress interface to use. In this example Egress Interface will be GIG 3 for CUCM.
    1. Refer to this document for CUBE call routing overview: https://www.cisco.com/c/en/us/support/docs/voice/ip-telephony-voice-over-ip-voip/211306-In-Depth-Explanation-of-Cisco-IOS-and-IO.html
5. CUBE will create a new TCP Socket and SIP INVITE all sourced from GIG 3 (INSIDE). CUBE operates in SELF zone so this will use the SELF-OUT zone-pair
6. The service-policy/policy-map **TRUSTED-POLICY-IN** will be used to inspect traffic based on **TRUSTED-CLASS-IN** class-map and **TRUSTED-ACL-IN** access-list
7. For return traffic in this flow **IN-SELF** and **SELF-OUT** zones to send responses for the call.

Show Commands

show zone-pair security

- This command will show all zone-pair mappings and the applied service policy.
- The source, destination keywords can be used to define a specific zone-pair mapping to check if many exist.

<#root>

```
Router#

show zone-pair security


Zone-pair name IN-SELF 2
    Source-Zone INSIDE Destination-Zone self
    service-policy TRUSTED-POLICY-IN
Zone-pair name OUT-SELF 4
    Source-Zone OUTSIDE Destination-Zone self
```

```
    service-policy TRUSTED-POLICY-OUT
Zone-pair name SELF-IN 5
    Source-Zone self Destination-Zone INSIDE
    service-policy TRUSTED-POLICY-IN
Zone-pair name SELF-OUT 6
    Source-Zone self Destination-Zone OUTSIDE
    service-policy TRUSTED-POLICY-OUT

Router#
```

`show zone-pair security source INSIDE destination self`

```
Zone-pair name IN-SELF 2
    Source-Zone INSIDE Destination-Zone self
    service-policy TRUSTED-POLICY-IN
```

show call active voice compact

- This command will show remote Media connections from the perspective of CUBE>

<#root>

```
Router#
```

`show call active voice com | i NA|VRF`

```
       <callID> A/O FAX   T<sec>  Codec       type        Peer Address   IP R:<ip>:<udp>              \
       467      ANS       T2      g711ulaw    VOIP        Psipp          192.168.1.48:16384
       468      ORG       T2      g711ulaw    VOIP        P8675309       192.168.3.59:16386           NA
```

show voip rtp connections

- This command shows both remote and local media connection information from the perspective of CUBE

<#root>

```
Router#
```

`show voip rtp con | i NA|VRF`

```
No. CallId     dstCallId  LocalRTP RmtRTP   LocalIP                                  RemoteIP
1     467        468        8120     16384    192.168.1.12                              192.168.1.48
2     468        467        8122     16386    192.168.2.58                             192.168.3.59
```

show call active voice brief

- This command, coupled with the media bulk-stats command configured via voice service voip will show send (TX) and received (RX) statistics for the call legs.
- If media is flowing through CUBE and ZBFW the TX should match the RX on a peer call leg. e.g 109 RX, 109 TX

<#root>

```
Router#
```

show call active voice br | i dur

```
 dur 00:00:03 tx:107/24156 rx:109/24592 dscp:0 media:0 audio tos:0xB8 video tos:0x0
 dur 00:00:03 tx:109/24592 rx:107/24156 dscp:0 media:0 audio tos:0xB8 video tos:0x0
```

show sip-ua connections tcp detail

- This command shows active SIP TCP connection details through CUBE
- Commands like **show sip-ua connections udp detail** or **show sip-ua connections tcp tls** detail can be used to show the same details for UDP SIP and TCP-TLS SIP

<#root>

```
Router#
```

show sip-ua connections tcp detail

```
Total active connections      : 2
[..truncated..]
Remote-Agent:192.168.3.52, Connections-Count:1
  Remote-Port Conn-Id Conn-State  WriteQ-Size            Local-Address               Tenant
  =========== ======= =========== =========== ===================================== ======
         5060      51 Established           0 192.168.2.58:51875                       0

Remote-Agent:192.168.1.48, Connections-Count:1
  Remote-Port Conn-Id Conn-State  WriteQ-Size            Local-Address               Tenant
  =========== ======= =========== =========== ===================================== ======
        33821      50 Established           0 192.168.1.12:5060                        0
[..truncated..]
```

show policy-firewall sessions platform

- This command will show the call from the ZBFW perspective.
- There will be SIP Sessions and sub-flows for RTP and RTCP.
- The session ID from this output can be used when debugging ZBFW later.
- **show policy-firewall sessions platform detail** can be used to view even more data.

<#root>

```
Router#
```

show policy-firewall sessions platform

```
--show platform hardware qfp active feature firewall datapath scb any any any any any all any --
[s=session  i=imprecise channel c=control channel  d=data channel u=utd inspect A/D=appfw action allow/d
Session ID:0x000000A8 192.168.2.58 51875 192.168.3.52 5060 proto 6 (-global-:0:-global-:0) (0x16:sip) [s
 +-Session ID:0x000000AA 192.168.2.58 0 192.168.3.52 5060 proto 6 (-global-:0:-global-:0) (0x16:sip)  [i
 +-Session ID:0x000000A9 192.168.3.52 0 192.168.2.58 5060 proto 6 (-global-:0:-global-:0) (0x16:sip)  [i
Session ID:0x000000AC 192.168.3.59 16386 192.168.2.58 8122 proto 17 (-global-:0:-global-:0) (0x2:udp) [s
Session ID:0x000000AD 192.168.1.48 16384 192.168.1.12 8120 proto 17 (-global-:0:-global-:0) (0x3a:sip rt
Session ID:0x000000A6 192.168.1.48 33821 192.168.1.12 5060 proto 6 (-global-:0:-global-:0) (0x16:sip)
```

```
 +-Session ID:0x000000AE 192.168.1.48 16385 192.168.1.12 8121 proto 17 (-global-:0:-global-:0) (0x3a:sip
 +-Session ID:0x000000AD 192.168.1.48 16384 192.168.1.12 8120 proto 17 (-global-:0:-global-:0) (0x3a:sip
 +-Session ID:0x000000AB 192.168.1.48 0 192.168.1.12 5060 proto 6 (-global-:0:-global-:0) (0x16:sip)
 +-Session ID:0x000000A7 192.168.1.12 0 192.168.1.48 5060 proto 6 (-global-:0:-global-:0) (0x16:sip)
```

show policy-map type inspect zone-pair sessions

- This command shows similar data as **show policy-firewall sessions platform** however the zone-pair mapping is also included in the output which is handy for debugging.

```
Router# show policy-map type inspect zone-pair sessions | i Zone-pair|Session ID
  Zone-pair: IN-SELF
         Session ID 0x000000AD (192.168.1.48:16384)=>(192.168.1.12:8120) sip-RTP-data SIS_OPEN
         Session ID 0x000000A6 (192.168.1.48:33821)=>(192.168.1.12:5060) sip SIS_OPEN
         Session ID 0x000000A7 (192.168.1.12:0)=>(192.168.1.48:5060) sip SIS_PREGEN
         Session ID 0x000000AE (192.168.1.48:16385)=>(192.168.1.12:8121) sip-RTP-data SIS_PREGEN
         Session ID 0x000000AB (192.168.1.48:0)=>(192.168.1.12:5060) sip SIS_PREGEN
  Zone-pair: OUT-SELF
         Session ID 0x000000AC (192.168.3.59:16386)=>(192.168.2.58:8122) udp SIS_OPEN
  Zone-pair: SELF-IN
  Zone-pair: SELF-OUT
         Session ID 0x000000A8 (192.168.2.58:51875)=>(192.168.3.52:5060) sip SIS_OPEN
         Session ID 0x000000AA (192.168.2.58:0)=>(192.168.3.52:5060) sip SIS_PREGEN
         Session ID 0x000000A9 (192.168.3.52:0)=>(192.168.2.58:5060) sip SIS_PREGEN
```

Troubleshoot

Troubleshooting Cisco IOS XE zone-based firewall can be found in this document:

https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/117721-technote-iosfirewall-00.html

CUBE Local Transcoding Interface (LTI) + ZBFW

- When CUBE is configured with hardware PVDM resources on the motherboard or a network interface module (NIM) these can be used for CUBE LTI purposes.
- The backplane interface for the PVDM will have a static service-engine x/y/z which corresponds to the placement of the PVDM. for example service-engine 0/4 is the motherboard PVDM/DSP slot.
- This service-engine MUST be configured with a zone and does not exist in the self zone.

The following configuration will map the service-engine used by CUBE LTI to the INSIDE zone for ZBFW purposes.

```
!
interface Service-Engine0/4/0
 zone-member security INSIDE
!
```

Similar logic for service-engine zone-pair mapping can be used for Hardware PVDM/DSP based SCCP Media Resources and the SCCP Bind Interface however this topic is outside the scope of this document.