Configuring service-redirection Using Group Policy Option

Information About ePBR and Group Policy Option

Beginning with Cisco NX-OS Release 10.5(1)F, users can redirect traffic flows between endpoints part of different Security-Groups. The redirection can happen through a single service function (as a firewall or a load-balancer) or through a chain of service functions. A given service function is built with one or more endpoints, representing the service devices performing such function. Traffic flows can be load-balanced across these service endpoints, while ensuring that both directions of traffic flow symmetrically use the same service endpoint. The onboarding of these service-devices, health monitoring mechanisms, and the user intent of chaining and load-balancing the traffic based on the properties of these service devices is captured and enforced through ePBR. To know more about micro-segmentation configuration, See Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

ePBR Service and Service-chain

You must first create a service function, which is defined with one or more endpoints with their specific attributes. Service endpoints are the service appliances such as firewall, IPS, and so on, that are available in the network to which traffic needs to be redirected. You can also define probes to monitor the health of the service endpoints. ePBR also supports load balancing along with service chaining. ePBR allows you to configure multiple active-active service endpoints as a part of a specific service function and would load-balance traffic among these endpoints.

You must specify the VRF context for the service as the context in which the endpoints are reachable.

After creating the ePBR service, you must create an ePBR service-chain. The ePBR service-chain allows you to define the service or services to which traffic should be redirected along with the order in which this needs to be done.

Services used in a chain are identified by a sequence number. In NXOS 10.5(1)F, only a single service function may be specified inside a service-chain, thereby supporting only redirection and load-balancing capabilities to a single service functions before traffic is permitted to its destination.

In every service sequence, you can define the fail-action method such as drop, forward, and bypass indicating the action that needs to be taken in the event of failures of all endpoints in the service. If no fail-action is configured, the default behavior is to drop the traffic when the service is considered as failed.

The ePBR service-chain also allows you to specify the manner in which traffic needs to be load-balanced amongst the endpoints inside a service.

Security Group for Service

You must configure security-group identifiers specific to the forward and reverse arms of ePBR services in order to use the service for micro-segmentation based redirection and chaining. This configuration is required to correctly steer the traffic to the service devices and through the chain.

These security-groups must be defined in the system as selector of type layer4-7. Each of the connected interfaces for the service endpoints inside the service must be mapped to the correct security-group as match interface selectors. For more details, see *Creating a Security Group on* Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

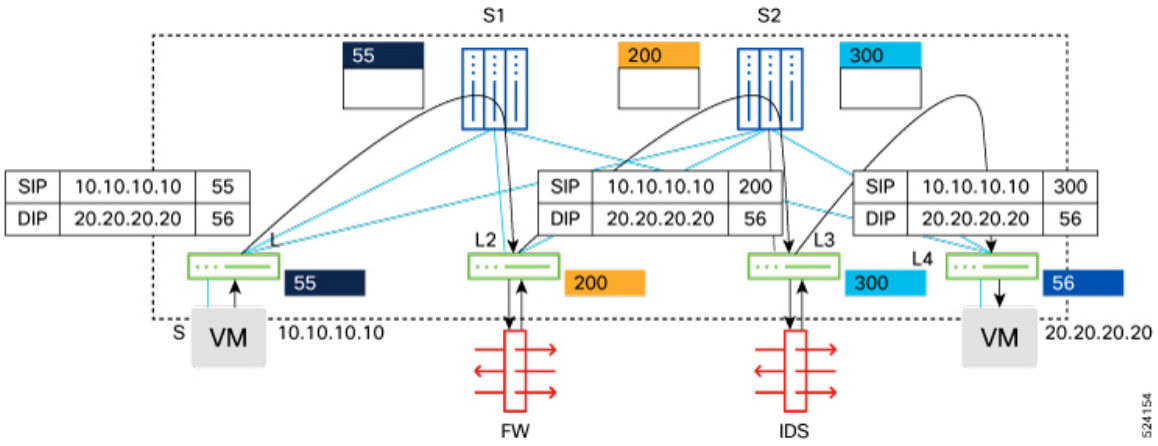
Connected interfaces for all the forward arms of the service endpoints must be mapped to the same identifier that is specified as the forward security-group for the ePBR service.

Connected interfaces for all the reverse arms of the service endpoints must be mapped to the same identifier that is specified as the reverse security-group for the ePBR service.

Only one security-group identifier should be configured for ePBR services with one-arm endpoints.

Two unique forward and reverse security-group identifiers should be configured for ePBR services with dual-arm endpoints. See figure 1 for a topology that explains the micro-segmentation based redirection and chaining.

Figure 1: Micro-segmentation with Service Chaining



Using ePBR Service-chains with SGACL Policies and Contracts

ePBR service-chain with micro-segmentation can provide traffic redirection using SGACL policies and contracts. Service-chain can be enabled for security contracts by attaching it to match class-maps inside policies used by contracts. For more details about the configuration, see Micro-segmentation for VXLAN Fabrics Using Group Policy Option (GPO).

ePBR Health Monitoring and Fail-action

ePBR monitors the health of the endpoints by provisioning IP SLA probes and object tracks to track the IP SLA reachability when you apply the probe configuration.

ePBR supports various probes for protocols such as ICMP, TCP, UDP, DNS, and HTTP. ePBR also supports user defined tracks, which allows you to create tracks with various parameters including millisecond probes and associate them with ePBR endpoints.

You can configure ePBR probe options for a service if all the endpoints of the service require similar probing methods and protocols. If one or more endpoints require a different probing mechanism, you can configure probe options specific to those forward and reverse endpoints. You can also configure frequency, timeout, retry up and down counts. For service-endpoints distributed in a VXLAN environment, you must configure source loopback interfaces for the probe, so that a unique source IP may be used for IP SLA sessions.

When probes are configured for the service, a unique loopback interface may be provided for the forward arms and the reverse arms of the service.

You can define tracks separately and assign the track ID to the forward and reverse arm of each service-endpoint in ePBR. These track IDs should not be re-used across different endpoints in the same or different ePBR service but may be shared between the forward and reverse arms of the endpoint. If you do not assign any user-defined track to an endpoint, ePBR will create a track using the probe method for the endpoint. If no probe method is defined at the endpoint level, the probe method configured for the service level will be used.

In events of device failures, traffic that was redirecting to the failed devices will redirect to other reachable devices, until the service is detected as failed. Resilient hash is supported during device failures. Traffic that was always being redirected to active service devices continues to redirect to the same devices in events of failures of other devices.

ePBR supports the following fail-action mechanisms for its service chain sequences:

- Drop
- Forward
- Bypasss

Drop of a service sequence indicates that the traffic must be dropped when the service in the current sequence is considered as failed. This is the default behavior when no fail-action is configured.

Forward indicates that upon failure of the service in the current sequence, traffic should use the regular routing. This fail-action mechanism is only supported when a single service function is available in the chain.

Bypass of a service sequence indicates that the traffic must be redirected to the next service sequence when the service in the current sequence is considered as failed. For a service-chain with a single sequence, traffic would use regular routing like the fail-action option of forward.

Load-Balancing Methods for Service Functions

Beginning with Cisco NX-OS 10.5(1)F, ePBR with micro-segmentation supports load-balancing traffic between service endpoints that are part of the same service function. Load-balance method may be configured for a service-chain if the same load-balance mechanism is desired for every service function in the chain. If one or more service functions or sequences inside the chain require a different load-balancing mechanism, this may be configured for the specific sequence inside the chain. Traffic may be load-balanced using source IP parameters, destination-IP parameters or source IP, destination IP along with the protocol indications available in the IP headers. ePBR with micro-segmentation ensures traffic is symmetrically load-balanced to the same service device in both directions.

Weighted Load-balancing

Beginning with Cisco NX-OS 10.5(1)F, ePBR with micro-segmentation supports load-balancing traffic to service endpoints proportional to the configured weights of the endpoints.

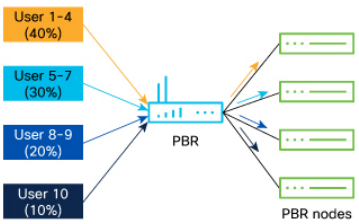
Each service device has weight configuration. The weight range is 1-10. The total number of weights per service is up to 128. With no weights configured, all service-endpoints configured inside an EPBR service today are treated as having a weight of 1, and traffic is load-balanced through equal-cost multipath mechanisms.

The service-endpoints inside an ePBR service may be optionally configured with weights that reflect the relative bandwidth or capacity of the devices, allowing more traffic to be redirected to devices with relatively higher weights and less traffic to be redirected to devices with relatively lower weights.

During endpoint failures, endpoints with higher weights will be preferred over endpoints with lower weights to receive the traffic of the failed endpoints.

Note that the weighted traffic distribution to the service devices is still dependent on the choice of the load-balancing algorithm and the distribution of the source and/or destination IP addresses of the traffic flows being received for service-chaining by the Nexus 9000 switch. See figure 2 for a weighted load-balancing arrangement.

Figure 2: Weighted Load-balancing



Destination	IP	Weight (optional)	%
Destination 1	10.1.1.1	4	40%
Destination 2	10.1.1.2	3	30%
Destination 3	10.1.1.3	2	20%
Destination 4	10.1.1.4	1	10%

524155

N+M Redundancy

Beginning with Cisco NX-OS 10.5(1)F, ePBR with micro-segmentation supports the ability to define service endpoints in hot-standby mode. M hot-standby service endpoints may be defined for a service function, with N active endpoints. When all primary endpoints are available, no traffic is redirected to hot-standby endpoints.

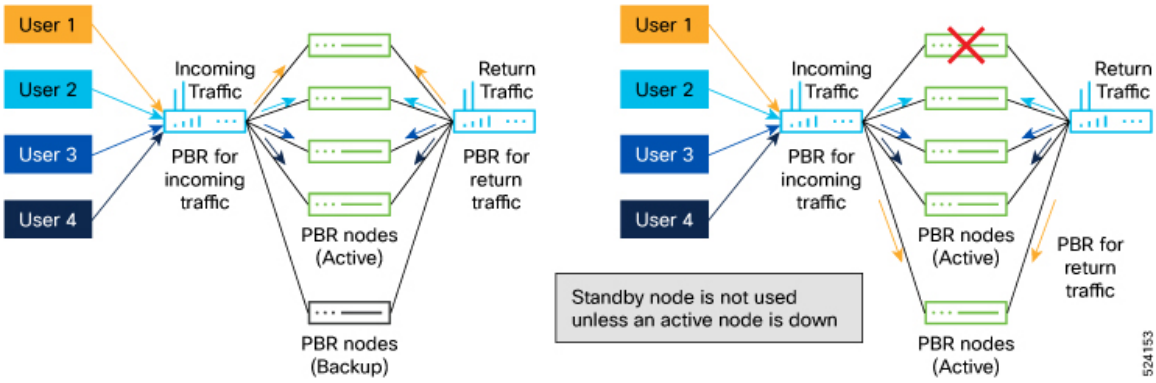
On failure of an active endpoint inside an ePBR service with hot-standby endpoints, traffic that was load-balanced to the failed endpoint, is now redirected to an available hot-standby endpoint.

On subsequent failures of more active endpoints and after all hot-standby endpoints have been utilized as backups for active endpoints, traffic from newly failed active endpoints may start redirecting to one or more available active and hot-standby endpoints.

When the active endpoint recovers, traffic that was being redirected to it, prior to its failure, will be restored to it. This behavior is unavoidable, and the traffic sessions may be required to get reestablished through the restored, stateful service endpoint.

Hot-standby endpoints may be configured with weights. On failure of a weighted active endpoint inside an ePBR service with weighted hot-standby endpoints, traffic is first redirected to a weighted hot-standby endpoint with equal or higher weight than the failed active endpoint. See figure 3 for a N+M redundancy arrangement.

Figure 3: N+M Redundancy



Redirection to NAT Devices

Beginning with Cisco NX-OS 10.5(1)F, ePBR with micro-segmentation supports redirection of traffic to service devices that modify the destination and/or source IP addresses of the traffic. These devices may be external load-balancers, NATTing firewalls and CGNAT devices.

Service devices may perform only destination NAT (load-balancers with SNAT disabled), only source NAT (CGNAT devices for return traffic) or both (load-balancers with SNAT enabled).

Traffic to devices such as external load-balancers performing destination NAT in the forward direction do not need policy-based redirection but need to be permitted.

Similarly, traffic in the reverse direction returning to devices such as external load-balancers or CGNAT devices, that have performed Source NAT in the forward direction, do not need policy-based redirection, but need to be permitted.

Traffic to devices such as external load-balancers that do not have source NAT enabled, require policy-based redirection for traffic in the reverse direction.

As described above, traffic to these services needs to be handled in different ways based on their NAT capabilities. Additionally, due to the modification of the IP addresses of the traffic by these appliances, the destination and/or source security-group tags may be different before and after redirection to these services. Handling these variances may ordinarily require complex asymmetric, uni-directional contracts.

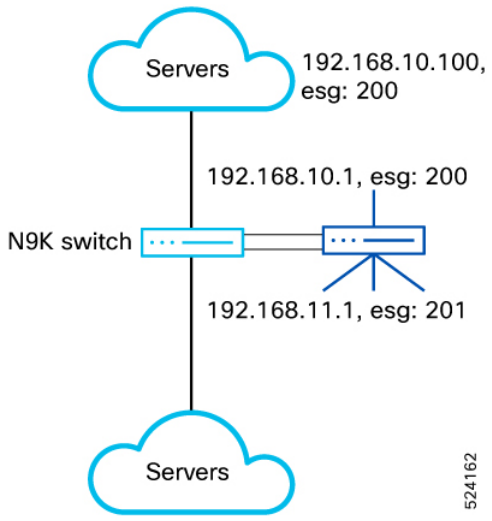
ePBR simplifies the contract creation for the user by allowing the user to indicate that the service in the ePBR service-chain at a particular sequence, has destination and/or source NAT capabilities. This is done by configuring an action for the service inside the chain, for the forward and reverse directions of traffic.

- Services that perform destination NAT on the traffic only are configured with action of route for the forward direction.
- Services that perform both destination and source NAT on the traffic, are configured with action of route for both directions of traffic.
- Services that perform only source NAT on the traffic are configured with action of route only for the reverse direction of traffic.

The user may then configure an end-to-end bi-directional contract for the traffic, regardless of any changes in security-tags in the traffic path and the varying behavior based on the direction.

When no action is configured for any direction, the service sequences inside the chain are treated as requiring redirection in both directions. Fail-action and threshold features will not be supported for sequences in the service-chain that have action of route configured for either the forward or reverse directions.

Figure 4: 2-arm Load-balancer (without SNAT) Service Device Insertion



Guidelines and Limitations

ePBR with micro-segmentation has the following guidelines and limitations:

- In NXOS 10.5(1)F SGACL based service redirection is only supported to a single service function in the chain. The service function may contain one or more layer-3 one-arm or dual-arm service endpoints only.
- For external load balancer, SGACL based service redirection only supports only one service endpoint in a service function and doesn't support load balance.
- Service functions with a mix of one-arm and dual-arm service endpoints are not supported.
- The sum total of weights across all active endpoints in the ePBR service cannot exceed 128.
- For the external load-balancer to monitor the health of the server cluster, contracts with permit action must be explicitly created between the layer4-7 security-tags of the load-balancer service and the servers.
- Service with one-arm devices must not be configured with a reverse security-group identifier.

- Services with dual-arm devices must be configured with a reverse security-group identifier that is different from the forward security-group.
- Services with dual-arm devices should use different service VLANs for the forward and reverse arms of the endpoints.
- The forward arms of one or more endpoints in the service may share the service VLAN. The reverse arms of one or more endpoints in the service may share the service VLAN.
- Users must ensure that service VLANs are used exclusively for the service devices and are not used for any other host traffic. This is required to avoid incorrect classification of such traffic.
- Security-groups configured inside ePBR services must also be defined as layer4-7 security-group selectors on the leaf switches which have the endpoint connected interfaces configured.
- In NXOS 10.5(1)F endpoint connected interfaces used in services must be interface VLANs only.
- While security-groups and service VLANs may be shared between ePBR services, users must ensure that the contracts that use these services in chains do not have conflicting match filters or actions.
- In NXOS 10.5(1)F, the service that the traffic is redirected to, must be configured in the same VRF context as the contract.
- Match class-maps for IPv4 traffic must be configured with service-chains containing IPv4 services and match class-maps for IPv6 traffic must be configured with service-chains containing IPv6 traffic.
- Unique layer-4 source and destination port parameters should be specified for the match class-map filters, if traffic is required to match any-any source and destination security-groups in the contract and the service in the chain is a dual-arm service.
- Users must ensure that multiple contracts using the same source and destination security-groups are not configured with policies and match class-maps having different service redirection results for the same traffic flows.
- When fail-action is configured for a sequence inside a service-chain , it is recommended that probing is consistently enabled for the service via service-level or endpoint-level probes.
- It is recommended that probe traffic is classified in a separate CoPP class. Otherwise, probe traffic may use the default CoPP class and might be dropped causing continuous IP SLA state changes during spikes in supervisor traffic. For information on CoPP configuration for IP SLA, see Configuring CoPP for IP SLA Packets.
- ePBR administrative and operational out-of-service features are not supported for services used in service redirection with micro-segmentation. For more information, see Configuring ePBR L3.
- Endpoint states of the forward and reverse arms of dual-arm devices are not synchronized automatically. If this is needed, identical probe track configuration on the forward and reverse arms should be used.
- Probe tracks configured for endpoints may be shared between the forward and reverse arms of the same endpoints, but not across endpoints in the same or different services.
- Probe tracks must be used for any automatic synchronization of endpoint states across the forward and reverse arms of dual-arm devices.

Configuring ePBR for Micro-segmentation

Configuring ePBR Service

Before you begin

The following section provides information about configuring ePBR services.

SUMMARY STEPS

1. **configure terminal**
2. **epbr service** *service-name*
3. **vrf** *vrf-name*
4. **[no] security-group <fwdGrp> [reverse**<*revGrp*>]
5. **[no] probe {icmp** | <*l4-proto*> **<port-num> [control**<*status*> **]** | **http get [** <*url-name*> **[version** <*ver*> **]** | **dns host** <*host-name*> **ctp} [frequency** <*freq-num*> | **timeout** <*timeout*> | **retry-down-count** <*down-count*> | **retry-up-count** <*up-count*> | **source-interface** <*src-intf*> | **reverse** <*rev-src-intf*> **]+**
6. **service-endpoint {ip** *ipv4-address* | **ipv6** *ipv6-address*}
7. **probe track** *track-ID*
8. **reverse {ip** *ipv4-address* | **ipv6** *ipv6-address*}
9. **mode hot-standby**
10. **weight** <*weight*>
11. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**epbr service** *service-name* **Example:** `switch(config)# epbr service firewall`	Creates a new ePBR service function.
Step 3	**vrf** *vrf-name* **Example:** `switch(config-epbr-svc)# vrf tenant_A`	Specifies the VRF for the ePBR service function.
Step 4	**[no] security-group <fwdGrp> [reverse**<*revGrp*>] **Example:**	Configures forward and reverse service security-group tags. For single arm devices, a single forward security-group must be specified. For dual arm devices the forward and reverse security-group must be unique.

	Command or Action	Purpose
	`switch(config-epbr-svc)# security-group 10 reverse 20` `switch(config-epbr-svc)# security-group 30`	The **no** form of this command removes the configuration.
Step 5	**[no] probe {icmp** | *<l4-proto>* **<port-num> [control***<status>* **] | http get [** *<url-name>* **[version** *<ver>* **] | dns host** *<host-name>* **ctp} [frequency** *<freq-num>* **| timeout** *<timeout>* **| retry-down-count** *<down-count>* **| retry-up-count** *<up-count>* **| source-interface** *<src-intf>* **| reverse** *<rev-src-intf>* **]+**	Configures the probe for the service function. The same configuration may also be applied for the forward and reverse arms of service endpoints. The no form of this command removes the configuration. For service-endpoints distributed in a VXLAN environment, you must configure source loopback interfaces for the probe, so that a unique source IP may be used for IP SLA sessions.
Step 6	**service-endpoint {ip** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `switch(config-vrf)# service-endpoint ip 172.16.1.200`	Configures service endpoint for the ePBR service. You can repeat steps 6 to 10 to configure another ePBR service endpoints.
Step 7	**probe track** *track-ID* **Example:** `switch(config-epbr-fwd-svc)# probe track 30`	Configures user-defined track for the forward or reverse arm of the service endpoint.
Step 8	**reverse {ip** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `switch(config-epbr-fwd-svc)# reverse ip 172.16.30.200`	Defines the reverse IP address for dual-arm service endpoints. Note that this is not needed for one-arm endpoints.
Step 9	**mode hot-standby** **Example:** `switch(config-epbr-fwd-svc)# mode hot-standby`	Configures the service-endpoint as a hot-standby endpoint.
Step 10	**weight** *<weight>* **Example:** `switch(config-epbr-fwd-svc)# weight 6`	Configures the weight for the active or hot-standby endpoint. Default value is 1.
Step 11	**exit** **Example:** `switch(config-vrf)# exit`	Exits the ePBR service configuration mode.

Configuring ePBR Service-chain

SUMMARY STEPS

1. **configure terminal**
2. **[no] epbr service-chain** *<chain-name>*
3. **load-balance method** *<lb-method>* **{ src-ip | dst-ip | src-dst-ipprotocol}**

4. *sequence-number* **set service** *service-name***[fail-action { bypass | drop | forward}]**
5. **action {route | redirect} [reverse-action {route| redirect}]**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**[no] epbr service-chain** <*chain-name*>	Configures ePBR service chain. The no form of this command removes the configuration.
Step 3	**load-balance method** <*lb-method*> **{ src-ip | dst-ip | src-dst-ipprotocol}** **Example:** `switch(config-epbr-svc-chain)# load-balance method src-ip`	Configures the load-balance method for the ePBR service-chain. The same configuration may also be applied to the individual service functions inside the service-chain. Default option is **src-dst-ipprotocol**.
Step 4	*sequence-number* **set service** *service-name***[fail-action { bypass | drop | forward}]** **Example:** `switch(config-epbr-svc-chain)# set service firewall fail-action drop`	Specifies the service function at the specific sequence in the chain and the fail-action mechanism for that sequence. Default option is **drop**.
Step 5	**action {route | redirect} [reverse-action {route| redirect}]** **Example:** `switch(config-epbr-svc-chain-seq)# action route reverse-action route`	Configure the forward and/or reverse action for the service in the chain to indicate destination and/or source NAT capabilities of the service. Default option is **redirect**.

Verifying ePBR Service-chain Configuration

Use the following commands to verify the ePBR service-chain configuration:

SUMMARY STEPS

1. **show epbr service [<svc-name>]**
2. **show epbr service-chain [<chain-name>] [reverse]**
3. **show tech-support epbr**
4. **show consistency-checker epbr service-chain { <svcChainName> | all }**
5. **show running-config epbr**
6. **show startup-config epbr**

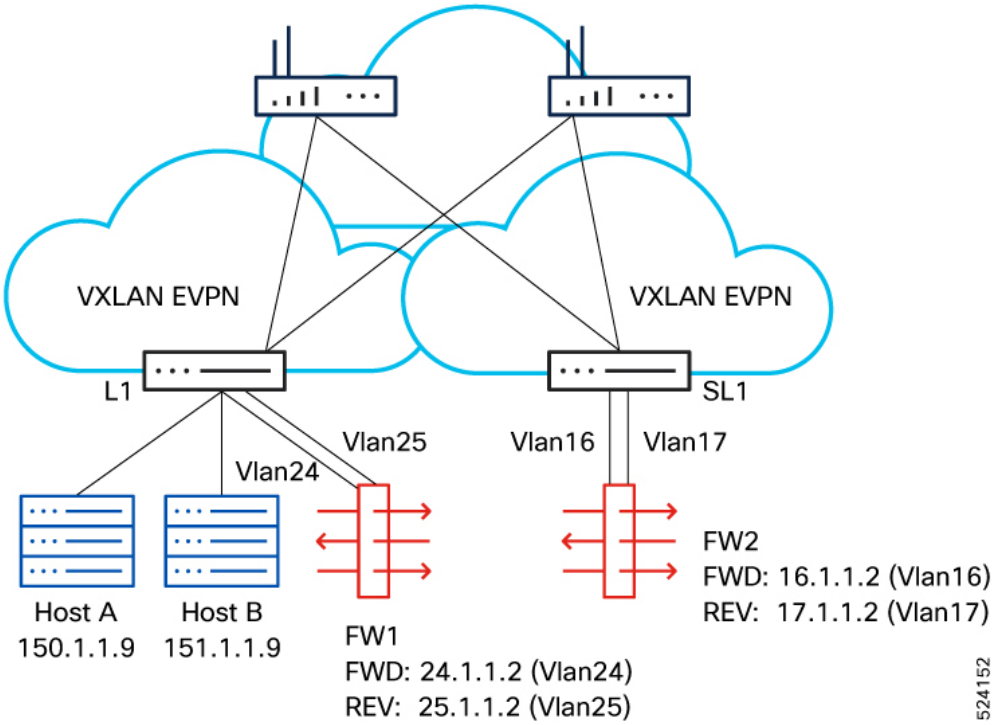
DETAILED STEPS

	Command or Action	Purpose
Step 1	**show epbr service [<svc-name>]** **Example:** `switch# show epbr service fw`	Displays information on the ePBR service function and endpoints.
Step 2	**show epbr service-chain [<chain-name>] [reverse]** **Example:** `switch# show epbr service-chain web`	Displays information on the ePBR service-chain in forward or reverse direction.
Step 3	**show tech-support epbr** **Example:** `switch# show tech-support epbr`	Displays the technical support information for ePBR.
Step 4	**show consistency-checker epbr service-chain { <svcChainName> | all }** **Example:** `show consistency-checker epbr service-chain web`	Performs consistency checks on ePBR configuration, redirection information for ePBR in the control plane and health monitoring mechanisms that are enabled.
Step 5	**show running-config epbr** **Example:** `switch# show running-config epbr`	Displays the running configuration for ePBR.
Step 6	**show startup-config epbr** **Example:** `switch# show startup-config epbr`	Displays the startup configuration for ePBR.

Configuration Examples for SGACL service-chaining Configuration

See figure 5 for the configuration example showing SGACL service-chaining configuration.

Figure 5: Configuration Example



1. Create layer4-7 selectors for the service.

```
security-group 2010 name fw_out
  type layer4-7
  match interface vlan 24
  match interface vlan 16
security-group 2011 name fw_in
  type layer4-7
  match interface vlan 25
  match interface vlan 17
```

2. Creating ePBR service and endpoints.

```
epbr service fw
  vrf tenant
  security-group 2010 reverse 2011
  probe tcp 80 frequency 5 timeout 3 source-interface
   loopback10 reverse loopback11
  service-end-point ip 24.1.1.2
    reverse ip 25.1.1.2
  service-end-point ip 16.1.1.2
    reverse ip 17.1.1.2
```

3. Create security-group selectors for host traffic.

```
security-group 5051 name sec_5051
  match connected-endpoints vrf tenant ipv4 151.1.1.0/24

security-group 5050 name sec_5050
  match connected-endpoints vrf tenant ipv4 150.1.1.0/24
```

4. Create security class-maps to define the layer3, layer-4 match criteria.

```
class-map type security match-any class_ipv4_tcp
  match ipv4 tcp dport 80
  match ipv4 tcp dport 443
```

5. Configure the ePBR service-chain. Configuration of class-maps, policy-maps and contracts under vrf need to be consistent on all leafs.

```
epbr service-chain web
  load-balance method src-dst-ipprotocol
  10 set service fw fail-action drop
```

6. Configure the security policy-map and attach the service-chain to the required match class-map.

```
policy-map type security web_policy
  class type security class_ipv4_tcp
    service-chain web
```

7. Configure the contract.

```
vrf context tenant
  security contract source 5050 destination 5051 policy web_policy
```

For more details on moving the VRF context to enforced mode, see *Configuring Security contracts between Security Groups*.

Verifying Configuration

- The following example shows how to verify ePBR service and endpoint.

```
show epbr service fw

Legend:

Operational State (Op-STS):  UP:Reachable, DOWN:Unreachable,

                             SVC-ADMIN-DOWN:Service shut

                             ADMIN-DOWN:Admin shut, OPER-DOWN:Out-of-service

Probe:
Protocol/Frequency(sec)/Timeout(sec)/Retry-Up-Count/Retry-Down-Count

Hold-down Threshold:         Count/Time(sec)

Service mode:                Full:Full-Duplex, Half:Half-Duplex

Type:                        L3:Layer-3, L2:Layer-2

Threshold:                   Threshold High/Low


Name                                 Type      Service mode  VRF

================================================================================================


fw                                   L3        Full
tenant


Security-group    Reverse security-group   Threshold
```

```
=====================================================

2010              2011


Endpoint IP/Intf         Track SLA       Op-ST          Probe                Hold-down
 Role Weight

Reverse IP/Intf          Track SLA       Op-ST          Probe

=============================================================================================



24.1.1.2/               1  20001     UP          TCP/5/3/0/0
       A     1

25.1.1.2/               3   20003     UP          TCP/5/3/0/0


16.1.1.2/               2   20002     UP             TCP/5/3/0/0
          A     1

17.1.1.2/               4   20004     UP             TCP/5/3/0/0
```

- The following example shows how to verify the ePBR service-chain in forward or reverse direction.

```
show epbr service-chain web

Service-chain : web

    service:fw, sequence:10, fail-action:Drop

      load-balance: Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 24.1.1.2 track 1 [UP]

      IP 16.1.1.2 track 2 [UP]


show epbr service-chain web reverse


Service-chain : web

    service:fw, sequence:10, fail-action:Drop

      load-balance: Source-Destination-ipprotocol, action:Redirect

      state:UP

      IP 25.1.1.2 track 3 [UP]

      IP 17.1.1.2 track 4 [UP]
```

- The following example shows how to verify consistency checker for service chain.

```
show consistency-checker epbr service-chain chain1
EPBR CC: Service Chain validation passed
show consistency-checker epbr service-chain all
EPBR CC: Service Chain validation passed
```