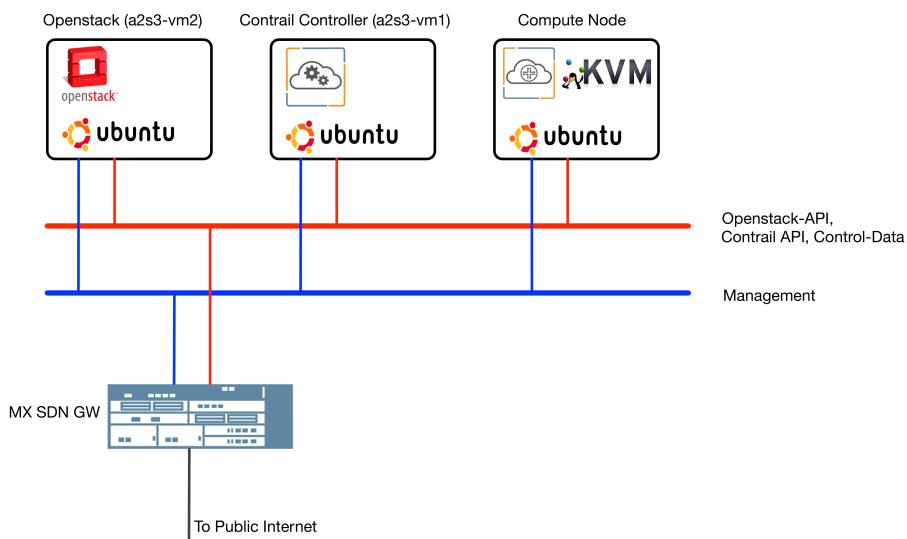


Contrail Service Chaining with PANW FW as the VNF

Vivekananda Shenoy (vshenoy@juniper.net)



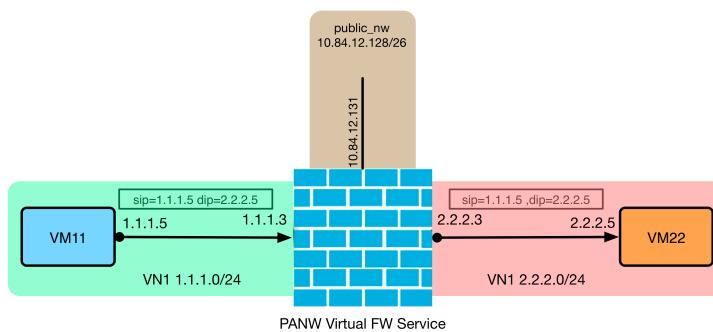
Physical Setup:



Use Case 1: PANW-FW as a Layer 3 firewall using Contrail In-Network service chaining feature.

In this use case PANW-FW is between the two Contrail Virtual Networks VN1 and VN2 and can inspect and filter the traffic that flows between VN1 and VN2 by making use of PANW-FW Zone based firewalls.

Logical Topology:



Configuration Steps:

Step1: Configure the Contrail Virtual Networks for LEFT, RIGHT and PUBLIC networks on the Contrail WebUI.

Note: Public network needs to be advertised to the MX-SDN GW for external internet since this is required for PANW-FW management WebUI (PANORAMA) access.

Network	Subnets	Attached Policies	Shared	Admin State
VN1	1.1.1.0/24	-	Disabled	Up
public_nw	10.84.12.128/26	-	Disabled	Up
VN2	2.2.2.0/24	-	Disabled	Up

Step2: Create the glance image for PANW FW VM.

```
glance image-create --disk-format qcow2 --file PA-VM-KVM-dhcp-enabled-7.1.4.qcow2 --is-public True --container-format bare --name panos-snapshot -progress
```

Step3: Launch the PANW FW VM using the above created image, nova flavor of m1.large and connected to 3 networks public_nw, VN1 and VN2. Make sure that the nova status shows that the VM is running.

```
nova boot --image 57cb1b06-ddeb-405b-bf19-6275c2f8261d --flavor m1.large --nic net-id=3af3760f-a323-4d81-aedc-cab9c0fb5e13 --nic net-id=f9fa95310886a364 --nic net-id=d6db9fb4-6609-44bf-bef2-bbd1a1432c98 --availability-zone nova:a2s37 PAN_FW
```

	Project	Host	Name	Image Name	IP Address	Size	Status	Task	Power State	Time since created	Actions
<input type="checkbox"/>	admin	a2s37	PAN_FW	panos-snapshot	public_nw 10.84.12.131 VN2 2.2.2.3 VN1 1.1.1.3	m1.large	Active	None	Running	21 hours, 8 minutes	<button>Edit Instance</button>

Step 4: Create the V2 service template of the type in-network firewall and virtual machine. Add 3 interfaces namely management, left and right to the service template.

Step 5: Create the v2 service instance using port tuple feature and add the 3 interfaces of the PANW-FW VM to the port-tuple.

Create

Name	Service Template								
PANW_SI	PAN_FW - [in-network (management, left, rig... ▾								
Interface Type									
management	public_nw								
left	VN1								
right	VN2								
Port Tuples									
Tuple + port-tuple0 : 10.84.12.131, 1.1.1.3, 2.2.2.3 + - <table border="1"> <tr> <td>Interface Type</td> <td>Virtual Machine Interface</td> </tr> <tr> <td>management</td> <td>(10.84.12.131) - 4211c1f9-0e20-...</td> </tr> <tr> <td>left</td> <td>(1.1.1.3) - 6d69bb64-b83e-4f3f-8...</td> </tr> <tr> <td>right</td> <td>(2.2.2.3) - 76b9ddd0-a604-4ccc-...</td> </tr> </table>		Interface Type	Virtual Machine Interface	management	(10.84.12.131) - 4211c1f9-0e20-...	left	(1.1.1.3) - 6d69bb64-b83e-4f3f-8...	right	(2.2.2.3) - 76b9ddd0-a604-4ccc-...
Interface Type	Virtual Machine Interface								
management	(10.84.12.131) - 4211c1f9-0e20-...								
left	(1.1.1.3) - 6d69bb64-b83e-4f3f-8...								
right	(2.2.2.3) - 76b9ddd0-a604-4ccc-...								

Cancel **Save**

Step 6: Configure a service policy between VN1 and VN2 to match all the application traffic and send it to the PANW_SI created in step 5.

Create

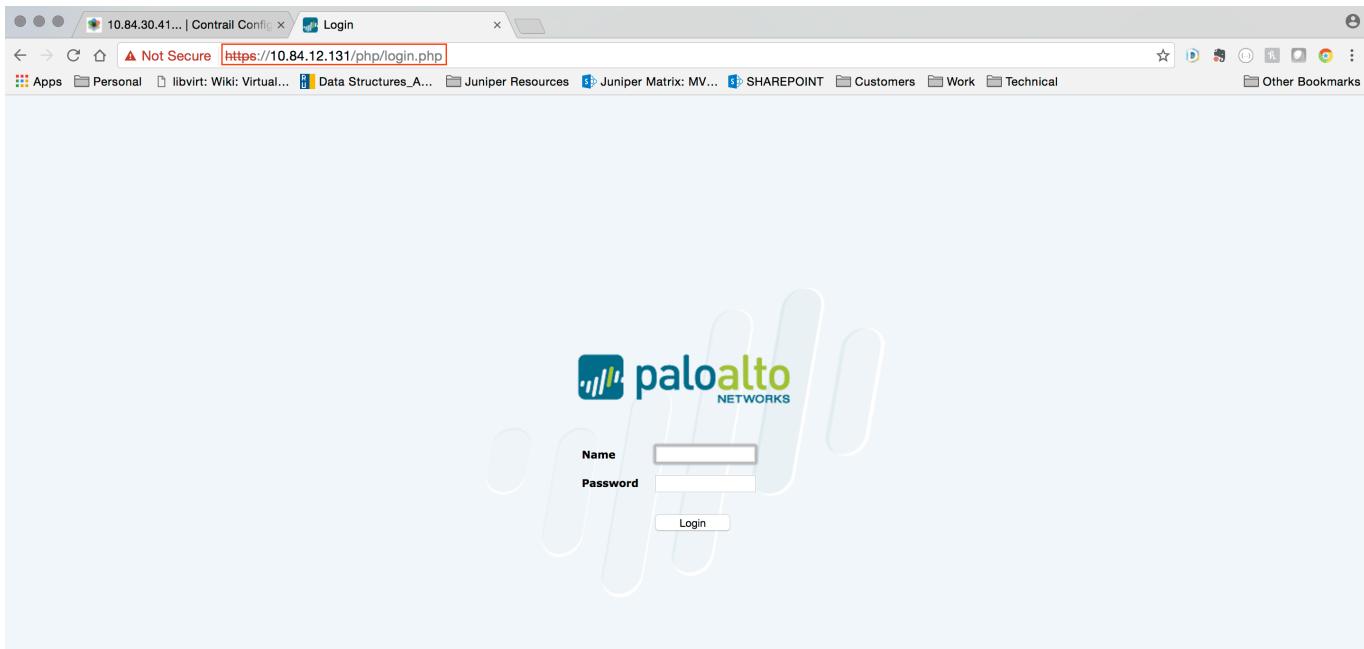
Policy	Permissions										
Policy Name											
FW_POL											
Policy Rule(s)											
Action	Protocol	Source	Ports	Direction	Destination	Ports	Log	Services	Mirror	QoS	+
PASS	ANY	VN1	ANY	<>	VN2	ANY	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	+ -
Service Instances											
PANW_SI											

Cancel **Save**

Step 7: Attach the FW_POL to VN1 and VN2.

Network	Subnets	Attached Policies	Shared	Admin State
VN1	1.1.1.0/24	FW_POL	Disabled	Up
public_nw	10.84.12.128/26	-	Disabled	Up
VN2	2.2.2.0/24	FW_POL	Disabled	Up

Step8: Login to PANW-FW Panorama management console by doing secure http to management IP address (10.84.12.131) using a web browser.



Step 9: Navigate to Network > Interface Management and create a management profile to permit all management / protocol traffic on the interfaces.

Name	Ping	Telnet	SSH	HTTP	HTTP OCSP	HTTPS	SNMP	Response Pages	User-ID	User-ID Syslog Listener-SSL	User-ID Syslog Listener-UDP	Permitted IP Addresses
all_traffic	<input checked="" type="checkbox"/>											

Step 10: Navigate to Network > Interfaces and configure the interfaces eth1/1 and eth1/2 which as per our topology is connected to Contrail virtual-networks VN1 and VN2 respectively. Select interface type as Layer3 and IP address as DHCP Client since our 2 Virtual Networks VN1 and VN2 have DHCP enabled.

Interface	Interface Type	Management Profile	Link State	IP Address	Virtual Router	Tag	VLAN / Virtual-Wire	Security Zone	Features	Comment
ethernet1/1	Layer3	all_traffic		Dynamic-DHCP Client	default	Untagged	none	none		
ethernet1/2	Layer3	all_traffic		Dynamic-DHCP Client	default	Untagged	none	none		
ethernet1/3				none	none	Untagged	none	none		
ethernet1/4				none	none	Untagged	none	none		
ethernet1/5				none	none	Untagged	none	none		
ethernet1/6				none	none	Untagged	none	none		
ethernet1/7				none	none	Untagged	none	none		

Step 10: Next navigate to Network > Zones and create 2 Security zones Left and Right of the type Layer 3. Attach the zone Left to eth1/1 and zone Right to eth1/2.

Name	Type	Interfaces / Virtual Systems	Zone Protection Profile	Log Setting	Enabled	User ID	Included Networks	Excluded Networks
LEFT	layer3	ethernet1/1			<input checked="" type="checkbox"/>	any	any	none
RIGHT	layer3	ethernet1/2			<input checked="" type="checkbox"/>	any	any	none

Step 11: Next navigate to Policies > Security and create a security policy between the LEFT and the RIGHT zones to permit all the traffic flowing from LEFT to RIGHT.

Name	Tags	Type	Zone	Address	User	HIP Profile	Zone	Address	Application	Service	Action
1 LEFT_RIGHT	none	interzone	LEFT	any	any	any	RIGHT	any	any	application-d...	Allow
2 intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any	Allow
3 interzone-default	none	interzone	any	any	any	any	any	any	any	any	Deny

Step 12: Final step is to commit the configuration so that all the above features we have configured on the PANW-FW takes effect.

Commit Status

Operation Commit
Status Active
Result Pending
Progress 99%

Verification:

Step 13: Launch 2 Ubuntu VM's in VN1 and VN2 which we will use to test the configuration.

Project	Host	Name	Image Name	IP Address	Size	Status	Task	Power State	Time since created	Actions
admin	a2s37	VM22	ubuntu	2.2.2.5	m1.medium	Active	None	Running	2 hours, 38 minutes	<button>Edit Instance</button>
admin	a2s37	VM11	ubuntu	1.1.1.5	m1.medium	Active	None	Running	2 hours, 38 minutes	<button>Edit Instance</button>
admin	a2s37	PAN_FW	panos-snapshot	public_nw 10.84.12.131 VN2 2.2.2.3 VN1 1.1.1.3	m1.large	Active	None	Running	22 hours, 17 minutes	<button>Edit Instance</button>

Step 14: Verify the working of the service chain by pinging the address of VM22 (2.2.2.5) from VM11.

Instance Details: VM11

```

Connected (unencrypted) to: QEMU (instance-000001ce)
Send CtrlAltDel
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu#
root@vm11:/home/ubuntu# ping 2.2.2.5
PING 2.2.2.5 (2.2.2.5) 56(84) bytes of data.
64 bytes from 2.2.2.5: icmp_seq=1 ttl=61 time=4.31 ms
64 bytes from 2.2.2.5: icmp_seq=2 ttl=61 time=1.99 ms
64 bytes from 2.2.2.5: icmp_seq=3 ttl=61 time=2.35 ms
64 bytes from 2.2.2.5: icmp_seq=4 ttl=61 time=2.14 ms
64 bytes from 2.2.2.5: icmp_seq=5 ttl=61 time=2.37 ms
64 bytes from 2.2.2.5: icmp_seq=6 ttl=61 time=2.07 ms
64 bytes from 2.2.2.5: icmp_seq=7 ttl=61 time=2.00 ms
64 bytes from 2.2.2.5: icmp_seq=8 ttl=61 time=1.76 ms
64 bytes from 2.2.2.5: icmp_seq=9 ttl=61 time=2.40 ms
64 bytes from 2.2.2.5: icmp_seq=10 ttl=61 time=1.92 ms
64 bytes from 2.2.2.5: icmp_seq=11 ttl=61 time=2.01 ms
64 bytes from 2.2.2.5: icmp_seq=12 ttl=61 time=1.96 ms
64 bytes from 2.2.2.5: icmp_seq=13 ttl=61 time=1.75 ms
64 bytes from 2.2.2.5: icmp_seq=14 ttl=61 time=2.17 ms
64 bytes from 2.2.2.5: icmp_seq=15 ttl=61 time=2.03 ms

```

Step 14: We can verify these flows by checking the security sessions on the PANW-FW VM and by checking the vrouter flows on the compute node.

```
[edit]
admin@PA-VM# run show session all

-----
ID      Application     State   Type Flag Src[Sport]/Zone/Proto (translated IP[Port])
Vsys
                               Dst[Dport]/Zone (translated IP[Port])

-----
1153    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[20]/RIGHT  (2.2.2.5[20])
1154    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[21]/RIGHT  (2.2.2.5[21])
1152    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[19]/RIGHT  (2.2.2.5[19])
1151    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[18]/RIGHT  (2.2.2.5[18])
1155    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[22]/RIGHT  (2.2.2.5[22])
1150    ping          ACTIVE  FLOW   1.1.1.5[7851]/LEFT/1  (1.1.1.5[7851])
vsys1
2.2.2.5[17]/RIGHT  (2.2.2.5[17])
[edit]
admin@PA-VM#
```

```

root@a2s37:~# flow -l --match "2.2.2.5"
Flow table(size 80609280, entries 629760)

Entries: Created 1180 Added 1177 Processed 1180 Used Overflow entries 0
(Created Flows/CPU: 65 55 81 74 92 383 17 1 15 5 13 29 50 35 20 77 38 49 4 4 15 38 14 6) (offlows 0)

Action:F=Forward, D=Drop N=NAT(S=SNAT, D=DNAT, Ps=SPAT, Pd=DPAT, L=Link Local Port)
Other:K(nh)=Key_Nexthop, S(nh)=RPF_Nexthop
Flags:E=Evicted, Ec=Evict Candidate, N>New Flow, M=Modified Dm=Delete Marked
TCP(r=reverse):S=SYN, F=FIN, R=RST, C=HalfClose, E=Established, D=Dead

Listing flows matching ([2.2.2.5]:*)

  Index      Source:Port/Destination:Port      Proto(V)
-----
  71596<=>503080      2.2.2.5:7851          1 (2->5)
                      1.1.1.5:0
(Gen: 1, K(nh):46, Action:F, Flags:, QOS:-1, S(nh):46, Stats:79/7742, SPort 64906 TTL 0)

  189056<=>435688      2.2.2.5:7851          1 (1->4)
                      1.1.1.5:0
(Gen: 1, K(nh):30, Action:F, Flags:, QOS:-1, S(nh):30, Stats:79/7742, SPort 62172 TTL 0)

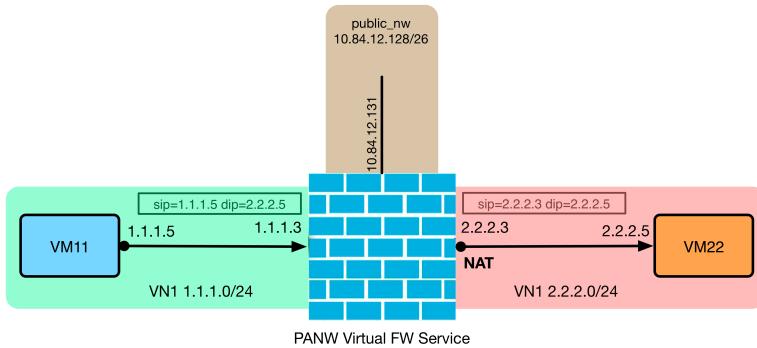
  435688<=>189056      1.1.1.5:7851          1 (1->4)
                      2.2.2.5:0
(Gen: 1, K(nh):42, Action:F, Flags:, QOS:-1, S(nh):42, Stats:79/7742, SPort 56755 TTL 0)

  503080<=>71596      1.1.1.5:7851          1 (2->5)
                      2.2.2.5:0
(Gen: 1, K(nh):34, Action:F, Flags:, QOS:-1, S(nh):34, Stats:79/7742, SPort 51356 TTL 0)

root@a2s37:~#

```

Use Case 2: PANW-FW as a Layer 3 firewall along with SNAT policy using Contrail In-Network NAT service chaining feature.



For this use case most of the configuration steps are same as the ones described in use case 1 with couple of differences.

1. On Contrail under Services > service template the service type should be configured as in-network-nat instead of in-network.
2. On PANW-FW a NAT policy has to be configured which should allow all the traffic from zone LEFT to zone RIGHT and the source address should be translated to interface IP address on eth1/2 (2.2.2.3) which is the interface in VN2. (Right Side Network).

Note: The Step 4", 5" and 6" replaces steps 4 , 5 & 6 from use case1. Step 11.1 is in addition to step 11 in use case1. All the other steps are same as the one shown in use case 1.

Step 4": On the Contrail Web-UI configure in-network-nat service template.

Step 5": Create the v2 service instance for the in-network-nat service template using port tuple feature and add the 3 interfaces of the PANW-FW VM to the port-tuple.

Step 6": Configure a service policy between VN1 and VN2 to match all the application traffic and send it to the PANW-FW-NAT service instance created in step 5".

Create

Policy Permissions

Policy Name: PANW_FW_NAT_POL

Policy Rule(s)

Action	Protocol	Source	Ports	Direction	Destination	Ports	Log	Services	Mirror	QoS	+
PASS	ANY	VN1	ANY	<>	VN2	ANY	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-

Service Instances: PANW-FW-NAT

Cancel Save

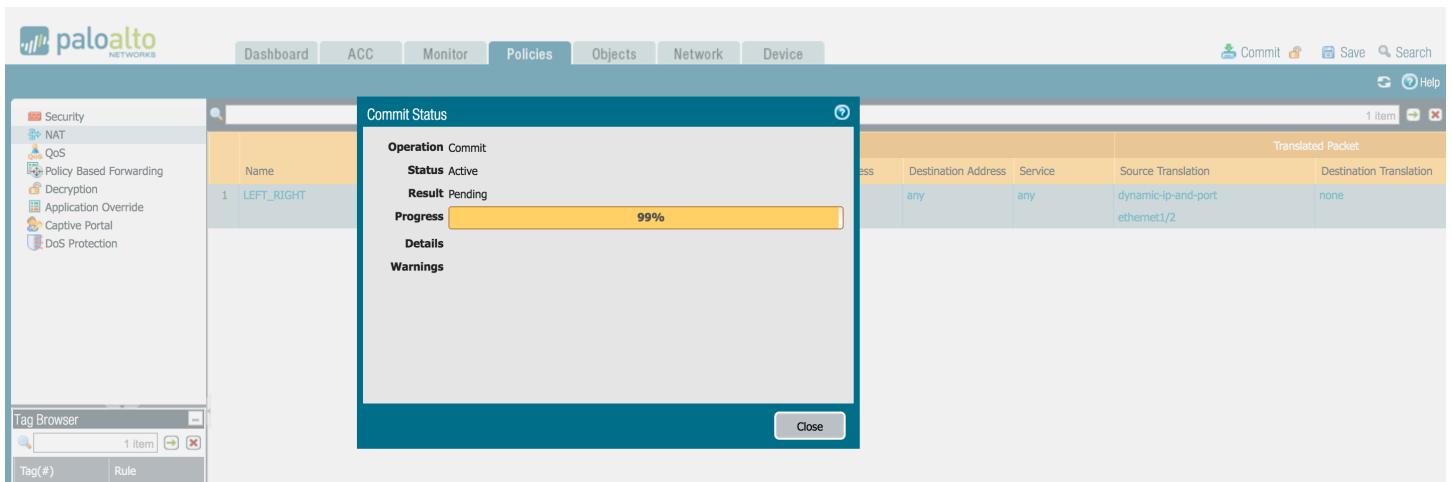
Step 7": Attach the service policy PANW_FW_NAT_POL to VN1 and VN2.

The screenshot shows the Juniper Junos Web Interface. The left sidebar is titled "Configure" and includes "Infrastructure", "Physical Devices", "Networking" (which is expanded to show "Networks", "Ports", "Policies", and "Security Groups"). The main content area is titled "Networks" and shows a table of subnets and their attached policies. The table has columns: Network, Subnets, Attached Policies, Shared, and Admin State. It lists three records: VN1 (1.1.1.0/24), public_nw (10.84.12.128/26), and VN2 (2.2.2.0/24). All three entries have "PANW_FW_NAT_POL" listed under "Attached Policies" and "Up" under "Admin State".

Step 11.1: On the PANW-FW VM configure a SNAT policy between the LEFT and RIGHT zones such that the source address is translated to eth1/2 interface address after SNAT.

The screenshot shows the Palo Alto Networks Global Protect UI. The top navigation bar includes "Dashboard", "ACC", "Monitor", "Policies" (which is selected and highlighted in blue), "Objects", "Network", and "Device". The left sidebar has a "Security" section with "NAT" selected, along with other options like "QoS", "Policy Based Forwarding", "Decryption", "Application Override", "Captive Portal", and "DoS Protection". The main content area displays a table for a NAT rule. The table has columns: Name, Tags, Source Zone, Destination Zone, Destination Interface, Source Address, Destination Address, Service, Source Translation, and Destination Translation. One row is shown with the following values: Name: LEFT_RIGHT, Tags: SNAT, Source Zone: LEFT, Destination Zone: RIGHT, Destination Interface: ethernet1/2, Source Address: any, Destination Address: any, Service: any, Source Translation: dynamic-ip-and-port, and Destination Translation: none.

Step 12": Commit the NAT policy on the PANW-FW VM.



Verification:

Step 14": Verify the working of the service chain by pinging the address of VM22 (2.2.2.5) from VM11. While ping is running login to VM22 and run tcpdump on the eth0 interfaces. The source IP address of the ICMP echo requests should be 2.2.2.3 which is the PANW-FW eth1/2 interface IP address. This shows that NAT is working.

Instance Details: VM11

Overview Log **Console** Action Log

Instance Console

If console is not responding to keyboard input: click the grey status bar below. [Click here to show only console](#)
To exit the fullscreen mode, click the browser's back button.

Connected (unencrypted) to: QEMU (instance-000000ce) Send CtrlAltDel

```
root@vm11:/home/ubuntu# ping 2.2.2.5
PING 2.2.2.5 (2.2.2.5) 56(84) bytes of data.
64 bytes from 2.2.2.5: icmp_seq=1 ttl=61 time=4.31 ms
64 bytes from 2.2.2.5: icmp_seq=2 ttl=61 time=1.99 ms
64 bytes from 2.2.2.5: icmp_seq=3 ttl=61 time=2.35 ms
64 bytes from 2.2.2.5: icmp_seq=4 ttl=61 time=2.14 ms
64 bytes from 2.2.2.5: icmp_seq=5 ttl=61 time=2.37 ms
64 bytes from 2.2.2.5: icmp_seq=6 ttl=61 time=2.07 ms
64 bytes from 2.2.2.5: icmp_seq=7 ttl=61 time=2.00 ms
64 bytes from 2.2.2.5: icmp_seq=8 ttl=61 time=1.76 ms
64 bytes from 2.2.2.5: icmp_seq=9 ttl=61 time=2.40 ms
64 bytes from 2.2.2.5: icmp_seq=10 ttl=61 time=1.92 ms
64 bytes from 2.2.2.5: icmp_seq=11 ttl=61 time=2.01 ms
64 bytes from 2.2.2.5: icmp_seq=12 ttl=61 time=1.96 ms
64 bytes from 2.2.2.5: icmp_seq=13 ttl=61 time=1.75 ms
64 bytes from 2.2.2.5: icmp_seq=14 ttl=61 time=2.17 ms
64 bytes from 2.2.2.5: icmp_seq=15 ttl=61 time=2.03 ms
```

Instance Details: VM22

Overview Log Console Action Log

Instance Console

If console is not responding to keyboard input: click the grey status bar below. [Click here to show only console](#)
To exit the Fullscreen mode, click the browser's back button.

```
Connected (unencrypted) to: QEMU (instance-000001cf) Send CtrlAltDel
root@vm22:/home/ubuntu#
root@vm22:/home/ubuntu# tcpdump -vv -c 4 -nei eth0 icmp
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
07:17:58.956661 02:76:b9:dd:d0:a6 > 02:29:c8:ff:d5:ae, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 48702, offset 0, flags [DF], proto ICMP (1), length 84)
    2.2.2.3 > 2.2.2.5: ICMP echo request, id 27921, seq 98, length 64
07:17:58.956709 02:29:c8:ff:d5:ae > 02:76:b9:dd:d0:a6, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 23428, offset 0, flags [none], proto ICMP (1), length 84)
    2.2.2.3 > 2.2.2.5: ICMP echo reply, id 27921, seq 98, length 64
07:17:59.958466 02:76:b9:dd:d0:a6 > 02:29:c8:ff:d5:ae, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 48703, offset 0, flags [DF], proto ICMP (1), length 84)
    2.2.2.3 > 2.2.2.5: ICMP echo request, id 27921, seq 99, length 64
07:17:59.958511 02:29:c8:ff:d5:ae > 02:76:b9:dd:d0:a6, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 23429, offset 0, flags [none], proto ICMP (1), length 84)
    2.2.2.5 > 2.2.2.3: ICMP echo reply, id 27921, seq 99, length 64
4 packets captured
4 packets received by filter
0 packets dropped by kernel
root@vm22:/home/ubuntu#
```

Step 14: We can also verify the NAT flows by checking the security sessions on the PANW-FW VM and by checking the vrouter flows on the compute node.

```
admin@PA-VM# run show session all

-----
ID      Application     State   Type Flag Src[Sport]/Zone/Proto (translated IP[Port])
Vsys          Dst[Dport]/Zone (translated IP[Port])
-----
1119      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[272]/RIGHT  (2.2.2.5[272])
1118      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[271]/RIGHT  (2.2.2.5[271])
1120      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[273]/RIGHT  (2.2.2.5[273])
1117      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[270]/RIGHT  (2.2.2.5[270])
1116      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[269]/RIGHT  (2.2.2.5[269])
1115      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[268]/RIGHT  (2.2.2.5[268])
1114      ping        ACTIVE  FLOW  NS  1.1.1.5[27921]/LEFT/1  (2.2.2.3[27921])
vsys1           2.2.2.5[267]/RIGHT  (2.2.2.5[267])
[edit]
admin@PA-VM#
```

```

root@a2s37:~# flow -l --match "2.2.2.5"
Flow table(size 80609280, entries 629760)

Entries: Created 8254 Added 8252 Processed 8254 Used Overflow entries 0
(Created Flows/CPU: 888 1036 1080 1147 1036 1794 17 1 15 5 13 29 156 186 207 182 156 224 4 4 16 38 14 6) (offlows 0)

Action:F=Forward, D=Drop N=NAT(S=SNAT, D=DNAT, Ps=SPAT, Pd=DPAT, L=Link Local Port)
Other:K(nh)=Key_Nexthop, S(nh)=RPF_Nexthop
Flags:E=Evicted, Ec=Evict Candidate, N>New Flow, M=Modified Dm=Delete Marked
TCP(r=reverse):S=SYN, F=FIN, R=RST, C=HalfClose, E=Established, D=Dead

Listing flows matching ([2.2.2.5]:*)

      Index          Source:Port/Destination:Port          Proto(V)
-----
286852<=>300324      1.1.1.5:27921                  1 (1->4)
                      2.2.2.5:0
(Gen: 1, K(nh):42, Action:F, Flags:, QOS:-1, S(nh):42, Stats:398/39004, SPort 62936 TTL 0)

300324<=>286852      2.2.2.5:27921                  1 (1->4)
                      1.1.1.5:0
(Gen: 1, K(nh):30, Action:F, Flags:, QOS:-1, S(nh):30, Stats:398/39004, SPort 63523 TTL 0)

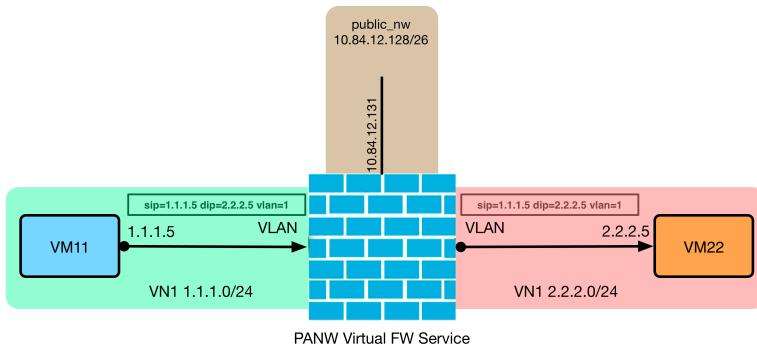
362872<=>478356      2.2.2.5:27921                  1 (2)
                      2.2.2.3:0
(Gen: 1, K(nh):46, Action:F, Flags:, QOS:-1, S(nh):46, Stats:398/39004, SPort 56227 TTL 0)

478356<=>362872      2.2.2.3:27921                  1 (2)
                      2.2.2.5:0
(Gen: 1, K(nh):34, Action:F, Flags:, QOS:-1, S(nh):34, Stats:398/39004, SPort 50299 TTL 0)

-----+-----+-----+-----+-----+-----+

```

Use Case 3: PANW-FW as a Layer 2 firewall (Virtual Wire) using Contrail transparent mode service chaining feature.



For this use case most of the configuration steps are same as the ones described in use case 1 with couple of differences.

1. On Contrail under Services > service template the service type should be configured as transparent mode instead of in-network.
2. On the PANW-FW for eth1/1 and eth1/2 the interface type should be configured as virtual-wire and these interfaces should be added to the virtual-wire VW1.

Note: The Step 4", 5" and 6" replaces steps 4, 5 & 6 from use case1. Step 11.1 is in addition to step 11 in use case1. All the other steps are same as the one shown in use case 1.

Step 4'': On the Contrail Web-UI configure transparent mode service template.

Create

Service Template Permissions

Name
PANW-FW-TRANSPARENT

Version
v2

Virtualization Type
Virtual Machine

Service Mode
Transparent

Service Type
Firewall

Interface(s)

- management
- left
- right

Cancel Save

Step 5'': Create the v2 service instance for the transparent mode service using port tuple feature and add the 3 interfaces of the PANW-FW VM to the port-tuple.

Create

Name
PANW-FW-TRANSPARENT-SI

Service Template
PANW-FW-TRANSPARENT - [transparent (ma...

Interface Type
Virtual Network

management	public_nw
left	VN1
right	VN2

Port Tuples

Tuple
port-tuple0 : 10.84.12.131, 1.1.1.3, 2.2.2.3

Interface Type management	(10.84.12.131) - 4211c1f9-0e20-...
left	(1.1.1.3) - 6d69bb64-b83e-4f3f-8...
right	(2.2.2.3) - 76b9ddd0-a604-4ccc...

Cancel Save

Step 6'': Configure a service policy between VN1 and VN2 to match all the application traffic and send it to the PANW-FW-TRANSPARENT-SI service instance created in step 5''.

The screenshot shows a 'Create' dialog for a 'Policy'. The 'Policy Name' field contains 'PANW-FW-TRANSPARENT-POL'. Under 'Policy Rule(s)', there is one rule with 'Action: PASS', 'Protocol: ANY', 'Source: VN1', 'Destination: ANY', and 'Service Instances: PANW-FW-TRANSPARENT-SI' selected. The 'Save' button is visible at the bottom right.

Step 7'': Attach the service policy PANW-FW-TRANSPARENT-POL to VN1 and VN2.

The screenshot shows the 'Configure' interface with 'Networking' selected. In the 'Networks' section, three entries are listed: 'VN1' (Subnet 1.1.0.0/24), 'public_nw' (Subnet 10.84.12.128/26), and 'VN2' (Subnet 2.2.2.0/24). Each entry has 'Attached Policies' set to 'PANW-FW-TRANSPARENT-POL' and 'Admin State' set to 'Up'. The 'Save' button is visible at the bottom right.

Step 10'': On the PANW-FW VM navigate to Network > Interfaces and configure the interface type for eth1/1 and eth1/2 as virtual wire from Layer3. Also under Network > Virtual Wires configure a virtual wire VW1 and add the 2 interfaces eth1/1 and eth1/2 to this virtual wire. Allow the complete range of vlan tags 1-4094.

The screenshot shows the Palo Alto Networks interface configuration. On the left, the navigation menu includes 'Interfaces', 'VLANs', 'Virtual Wires', and 'Virtual Wires'. In the main area, under 'Network > Virtual Wires', a new virtual wire 'VW1' is being created. The table shows two interfaces, 'ethernet1/1' and 'ethernet1/2', both configured as 'Virtual Wire' type. They are both associated with 'VW1' and have 'Untagged' as their tag type. The 'Comment' column is empty.

Name	Interface1	Interface2	Tag Allowed	Multicast Firewalling	Link State Pass Through
VW1	ethernet1/1	ethernet1/2	1-4094	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Step 11”: Under Network > Zones configure the type for LEFT and RIGHT zones as virtual-wire and add the eth1/1 and eth1/2 interfaces to LEFT and RIGHT zones respectively. Also configure a security policy to allow all the application traffic between the LEFT and the RIGHT zones.

Name	Type	Interfaces / Virtual Systems	Zone Protection Profile	Log Setting	Enabled	Included Networks	Excluded Networks
LEFT	virtual-wire	ethernet1/1			<input type="checkbox"/>	any	none
RIGHT	virtual-wire	ethernet1/2			<input type="checkbox"/>	any	none

Name	Tags	Type	Zone	Address	User	HIP Profile	Zone	Address	Application	Service	Action
1 LEFT-RIGHT	TRANSPARENT FW	universal	LEFT	any	any	any	RIGHT	any	any	application-d...	Allow
2 intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	any	any	Allow
3 interzone-default	none	interzone	any	any	any	any	any	any	any	any	Deny

Verification:

Step 14”: Verify the working of the service chain by pinging the address of VM22 (2.2.2.5) from VM11. While ping is in progress if we run tcpdump on tap interface that corresponds to eth1/1 or eth1/2 interface of PANW-FW we should see that the ICMP packets should be sent with vlan tags. This is the vlan tag allocated by vrouter for this transparent service. We can also verify the vrouter flows and the security session on the PANW-FW VM.

```
Connected (unencrypted) to: QEMU (instance-000001ce)
root@vm11:/home/ubuntu# ping 2.2.2.5
PING 2.2.2.5 (2.2.2.5) 56(84) bytes of data.
64 bytes from 2.2.2.5: icmp_seq=1 ttl=62 time=3.47 ms
64 bytes from 2.2.2.5: icmp_seq=2 ttl=62 time=1.62 ms
64 bytes from 2.2.2.5: icmp_seq=3 ttl=62 time=1.90 ms
64 bytes from 2.2.2.5: icmp_seq=4 ttl=62 time=1.80 ms
64 bytes from 2.2.2.5: icmp_seq=5 ttl=62 time=2.22 ms
64 bytes from 2.2.2.5: icmp_seq=6 ttl=62 time=1.74 ms
64 bytes from 2.2.2.5: icmp_seq=7 ttl=62 time=1.97 ms
64 bytes from 2.2.2.5: icmp_seq=8 ttl=62 time=2.19 ms
64 bytes from 2.2.2.5: icmp_seq=9 ttl=62 time=2.00 ms
```

```
root@a2s37:~# flow -l --match "2.2.2.5"
Flow table(size 80609280, entries 629760)

Entries: Created 40421 Added 40420 Processed 40421 Used Overflow entries 0
(Created Flows/CPU: 5957 5615 6046 6674 5993 7183 17 1 15 5 13 29 302 407 745 498 436 403 4 4 16 38 14 6) (offlows 0)

Action:F=Forward, D=Drop N=NAT(S=SNAT, D=DNAT, Ps=SPAT, Pd=DPAT, L=Link Local Port)
Other:K(nh)=Key_Nexthop, S(nh)=RPF_Nexthop
Flags:E=Evicted, Ec=Evict Candidate, N>New Flow, M=Modified Dm=Delete Marked
TCP(r=reverse):S=SYN, F=FIN, R=RST, C=HalfClose, E=Established, D=Dead

Listing flows matching ([2.2.2.5]:*)
-----
```

Index	Source:Port/Destination:Port	Proto(V)
60284<=>500420	2.2.2.5:27952 1.1.1.5:0	1 (2->5)
(Gen: 2, K(nh):46, Action:F, Flags:, QOS:-1, S(nh):46, Stats:2181/213738, SPort 64751 TTL 0)		
106176<=>153216	1.1.1.5:27952 2.2.2.5:0	1 (1->4)
(Gen: 215, K(nh):42, Action:F, Flags:, QOS:-1, S(nh):42, Stats:2181/213738, SPort 62001 TTL 0)		
153216<=>106176	2.2.2.5:27952 1.1.1.5:0	1 (4->4)
(Gen: 3, K(nh):36, Action:F, Flags:, QOS:-1, S(nh):36, Stats:2181/222462, SPort 59362 TTL 0)		
204956<=>495968	2.2.2.5:27956 1.1.1.5:0	1 (2->5)
(Gen: 2, K(nh):46, Action:F, Flags:, QOS:-1, S(nh):46, Stats:58/5684, SPort 63554 TTL 0)		

```
admin@PA-VM# run show session all
```

ID	Application	State	Type Flag	Src[Sport]/Zone/Proto (translated IP[Port])	Dst[Dport]/Zone (translated IP[Port])
Vsys					
213	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[111]/RIGHT (2.2.2.5[111])	
209	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[107]/RIGHT (2.2.2.5[107])	
211	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[109]/RIGHT (2.2.2.5[109])	
208	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[106]/RIGHT (2.2.2.5[106])	
210	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[108]/RIGHT (2.2.2.5[108])	
212	ping	ACTIVE	FLOW	1.1.1.5[27956]/LEFT/1 (1.1.1.5[27956])	
vsys1				2.2.2.5[110]/RIGHT (2.2.2.5[110])	
[edit]					
admin@PA-VM#					

```
root@a2s37:~# tcpdump -vv -c 8 -nei tap76b9ddd0-a6
tcpdump: WARNING: tap76b9ddd0-a6: no IPv4 address assigned
tcpdump: listening on tap76b9ddd0-a6, link-type EN10MB (Ethernet), capture size 65535 bytes
09:52:29.971816 02:00:00:00:00:01 > 02:00:00:00:00:02, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 21129, offset)
  1.1.1.5 > 2.2.2.5: ICMP echo request, id 27956, seq 179, length 64
09:52:29.972092 02:00:00:00:00:02 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 27710, offset)
  2.2.2.5 > 1.1.1.5: ICMP echo reply, id 27956, seq 179, length 64
09:52:30.973104 02:00:00:00:00:01 > 02:00:00:00:00:02, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 21130, offset)
  1.1.1.5 > 2.2.2.5: ICMP echo request, id 27956, seq 180, length 64
09:52:30.973408 02:00:00:00:00:02 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 27711, offset)
  2.2.2.5 > 1.1.1.5: ICMP echo reply, id 27956, seq 180, length 64
09:52:31.974302 02:00:00:00:00:01 > 02:00:00:00:00:02, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 21131, offset)
  1.1.1.5 > 2.2.2.5: ICMP echo request, id 27956, seq 181, length 64
09:52:31.974588 02:00:00:00:00:02 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 27712, offset)
  2.2.2.5 > 1.1.1.5: ICMP echo reply, id 27956, seq 181, length 64
09:52:32.975506 02:00:00:00:00:01 > 02:00:00:00:00:02, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 21132, offset)
  1.1.1.5 > 2.2.2.5: ICMP echo request, id 27956, seq 182, length 64
09:52:32.975772 02:00:00:00:00:02 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 102: vlan 1, p 0, ethertype IPv4, (tos 0x0, ttl 63, id 27713, offset)
  2.2.2.5 > 1.1.1.5: ICMP echo reply, id 27956, seq 182, length 64
8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@a2s37:~#
```