How To Configure Affinities and Firewall Instances on VSX Gateways

Objective
This document explains the steps needed to configure affinities and firewall instances on R77.20 VSX gateways running Gaia.

Details

Supported Versions
R77
R77.20
R77.30

Supported OS
Gaia

Before you start

Related Documentation and Assumed Knowledge
• R77 VSX Check Point Administration Guide
• R77 Check Point Performance Tuning Administration Guide

Warnings
• When the number of firewall instances on a Virtual System changes, there is some downtime for that Virtual System.
• Each firewall instance that you create uses additional system memory. A Virtual System with five instances would use approximately the same amount of memory as five separate Virtual Systems.
• We recommend that you keep the number of firewall instances on the Virtual System lower than, or the same as, the number of CPU cores assigned to the same Virtual System.

Lab Environment
For the purposes of this document, we created an R77.20 VSX gateway with 16 cores and 9 physical interfaces. The gateway runs Performance Pack and we manage it with an R77.20 Management Console.
Before Setting Affinities

If Performance Pack (software acceleration product) is running on a multi-core system, you must set affinities manually:

```
# sim affinity -s
```

`sim affinity` commands take effect only if Performance Pack is enabled and running. Performance Pack begins running when you install the Security Policy for the first time.

To see if Performance Pack is enabled and running:

```
# fwaccel stat
```
We recommend that you dedicate one processing core to each interface when possible. If there are more interfaces than cores, one or more cores handle two interfaces. Use interface pairs of the same position with internal and external bonds.

If you use interface bonds, you must define the bonds before you set affinities.

For example, we run `show interfaces` before the policy is installed. The output shows that the gateway has nine physical interfaces and two bond interfaces.

```
VSX-GATEWAY> show interfaces
bond1
bond2
eth0
eth1
eth2
eth3
eth4
eth5
eth6
eth7
eth8
lo
VXSGATEWAY>
```

We run `show bonding group 1`. The output shows that this is a bond of eth5 and eth6.

```
VXSGATEWAY> show bonding group 1
Bond Interfaces
  eth5
  eth6
```

Bond2 includes eth7 and eth8:

```
VXSGATEWAY> show bonding group 2
Bond Interfaces
  eth7
  eth8
VXSGATEWAY>
```
The interface list changes after we add two Virtual Systems and install the policy.

Interfaces **eth5** and **eth6** are not in the output because they are part of bond1, which does not have a VLAN Trunk defined. Therefore, bond1 does not belong to VS0 and its interfaces are not displayed in VS0 context.
Setting Affinities

Note - Affinities are set for each physical interface, not for each bond.

1. Log in to expert mode.
2. Run: # sim affinity -s
3. When prompted, enter the CPU core for each interface.

In the example, cpu0 is assigned to eth0.

Warnings about interrupt conflicts show, because of VMware constraints. Interfaces share the same IRQ (2 interfaces for each IRQ), and the same IRQ will share the same CPU core.

To see interface affinities, run: # sim affinity -l
To see interface affinities with IRQ, run: `# sim affinity -l -v`

```
[Expert@VSX-GATEWAY:0]# fw ctrl affinity -l -v
Interface eth0 (irq 67): CPU 0
Interface eth2 (irq 83): CPU 2
Interface eth3 (irq 59): CPU 3
Interface eth4 (irq 67): CPU 0
Interface eth7 (irq 59): CPU 3
Interface eth8 (irq 67): CPU 0
VS_0 fwk: CPU 2 3 4 5 6 7 8 9 10 11 12 13 14 15
VS_1 fwk: CPU 2 3 4 6 7 8 9 10 11 12 13 14 15
VS_2 fwk: CPU 2 3 4 6 7 8 9 10 11 12 13 14 15
VS_3 fwk: CPU 2 3 4 6 7 8 9 10 11 12 13 14 15
```

**Configuring CoreXL on Virtual Systems**

Firewall instances are configured differently for the VSX Gateway (VS0) than for Virtual Systems.

- **On VSX Gateway** - Use the CLI (cpconfig) to configure the number of instances.
- **On Virtual Systems** - Use SmartDashboard (Virtual System properties) to configure the number of instances.

In this document, we will create firewall instances for VS_1 and VS_2 Virtual Systems.

```
[Expert@VSX-GATEWAY:0]# vsx stat -v
VX Gateway Status
------------------
Name: VSX-GATEWAY
Security Policy: Standard
Installed at: 14Jan2015 19:11:17
SIC Status: Trust

Number of Virtual Systems allowed by license: 25
Virtual Systems [active / configured]: 2 / 2
Virtual Routers and Switches [active / configured]: 1 / 1
Total connections [current / limit]: 8 / 45600

Virtual Devices Status
-----------------------

<table>
<thead>
<tr>
<th>ID</th>
<th>Type &amp; Name</th>
<th>Security Policy</th>
<th>Installed at</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>S VS_1</td>
<td>Standard</td>
<td>14Jan2015 19:11:17</td>
</tr>
<tr>
<td>3</td>
<td>S VS_2</td>
<td>Standard</td>
<td>14Jan2015 19:11:17</td>
</tr>
</tbody>
</table>

Type: S - Virtual System, B - Virtual System in Bridge mode, R - Virtual Router, W - Virtual Switch.
```

To enable CoreXL firewall instances on a Virtual System:

1. Open SmartDashboard.
2. Open the Virtual System Properties (double-click the Virtual System object).
3. Click **CoreXL**.
4. In **Number of virtual system instances**, enter the number of firewall instances.
In this example, we configure two firewall instances for VS_1 and two for VS_2.

You can set up to ten firewall instances on each Virtual System. Each instance uses more system memory. Virtual Systems with five instances would use approximately the same amount of memory as five separate Virtual Systems.

To see CoreXL status:

1. Change environment to Virtual System VS_1:  
   `vsenv 2`
   where 2 is the Virtual System ID.
2. Run:  
   `# fw ctl multik stat`

In the example, we see that there are two firewall instances enabled on VS_1: ID 0 and ID 1.

By default, each instance has access to 14 CPU cores: CPU 2 to CPU 15.
Configuring Virtual System Affinities

To see Virtual System affinities:

1. Log in to expert mode.
2. Change environment to VS0: `# vsenv 0`
   The command to see affinities runs only from VS0.
3. Run: `# fw ctl affinity -l -v`

In our example, we see that each Virtual System has access to 14 CPU cores (CPU 2 to CPU 15).

To set affinities for Virtual Systems or processes:

1. Take note of the Virtual System IDs. Run: `> show virtual-system all`

2. Log in to expert mode.
3. Run: `# fw ctl affinity -s`

Notes:

- You can run `fw ctl affinity -s` from any environment.
- To save CoreXL affinities after reboot, use the `-d` flag.
- To set affinity for a specific Virtual System, use: `--vsid <VS ID>`

To set affinity for a Virtual System:

`# fw ctl affinity -s -d --vsid <VS ID> --cpu <CPU ID...CP ID>`
In our example, we set affinities for VS_1 and VS_2:

```
# fw ctl affinity -s -d -vsid 2 -cpu 7 8 9 10
# fw ctl affinity -s -d -vsid 3 -cpu 11 12 13 14
```

```
[Expert@VSX-GATEWAY:0]# fw ctl affinity -s -d -vsid 2 -cpu 7 8 9 10
VDevice 2 : CPU 7 8 9 10 - set successfully
[Expert@VSX-GATEWAY:0]#

[Expert@VSX-GATEWAY:0]# fw ctl affinity -s -d -vsid 3 -cpu 11 12 13 14
VDevice 3 : CPU 11 12 13 14 - set successfully
[Expert@VSX-GATEWAY:0]#
```

We set affinities for VSX (ID 0) and Virtual Switch (ID 1):

```
# fw ctl affinity -s -d -vsid 0 -cpu 5
# fw ctl affinity -s -d -vsid 1 -cpu 6
```

```
[Expert@VSX-GATEWAY:0]#
[Expert@VSX-GATEWAY:0]# fw ctl affinity -s -d -vsid 0 -cpu 5
VDevice 0 : CPU 5 - set successfully
[Expert@VSX-GATEWAY:0]# fw ctl affinity -s -d -vsid 1 -cpu 6
VDevice 1 : CPU 6 - set successfully
[Expert@VSX-GATEWAY:0]#
```

To see the current affinities, change environment to VS0 and run:

```
# fw ctl affinity -l
```

```
[Expert@VSX-GATEWAY:0]#
[Expert@VSX-GATEWAY:0]# fw ctl affinity -l
eth0: CPU 0
eth2: CPU 2
eth3: CPU 3
eth4: CPU 0
eth7: CPU 3
eth8: CPU 0

VS_0: CPU 5
VS_1: CPU 6
VS_2: CPU 7 8 9 10
VS_3: CPU 11 12 13 14
```

Note – The number of firewall instances configured on a Virtual System does not have to match the number of CPU cores assigned to the Virtual System, but we do not recommend to have more firewall instances than CPU cores on the same Virtual System.
Verification

Monitor configured affinities on a VSX gateway:

1. Change environment to VS0.
2. Run: 
   ```
   # fw ctl affinities -l
   ```

Monitor affinities of the processes on the VSX gateway:

   ```
   # fw ctl affinity -l -x
   ```

Monitor affinities for the processes on a specific Virtual System.

   ```
   # fw ctl affinity -l -x -vsid <VS ID> -flags tn
   ```

For example, with

   ```
   # fw ctl affinity -l -x -vsid 2 -flags tn
   ```

we see that all processes on VSID 2 (Virtual System VS_1) are running on CPU cores 7 8 9 and 10.