puresecurity™

Check Point SOFTWARE TECHNOLOGIES LTD.

Performance Tests Methodology

Revised: June 15, 2007

This document provides configuration information for best performance results.

In This Document

Maximum Concurrent Connections

- Testing numerous connections may cause you to reach the concurrent connections limit. If this occurs new connections will fail.
- To prevent this scenario, verify that the concurrent connections do not exceed the limit of the system that is dependant on the following:
  - the maximum concurrent connections configured in the policy
  - the system's memory capacity
- Assess how many concurrent connections occur during the test load and change the maximum concurrent connections in the policy accordingly.

  To change the maximum concurrent connections perform the following:

  1. Select the object.
  2. Right click and select **Edit**.
  3. In the window the appears select the **Capacity Optimization** tab.
  4. Change the **Maximum concurrent connections**.

  For example, if TCP end timeout (defined in the policy's **Global Properties** window) is set to 20 seconds, and the session rate is approximately 20,000 Session/Sec, the number of concurrent connections are approximately 400,000.
- Verify if the concurrent connections of the test have reached the max concurrent connections configured in the policy as follows:
  1. Execute `fw tab -t connections -s` after the test is complete.
  2. Compare the peak concurrent connections with the maximum concurrent connections configured in the policy.
- In some cases, the maximum concurrent connections is configured high enough to satisfy the test requirements. However, the memory capacity of the system may limit the concurrent connections capacity and as a result it will not allow new connections to be created.

  To detect whether or not the system memory capacity was reached, execute `fw ctl pstat` and check for failed allocations in the **System kernel memory (smem) statistics** section.
- If the new connections are not created because the system reached its limits, you should reduce the concurrent connections number by reducing the TCP end timeout. Changing the TCP end timeout can be done in the **Global Properties** in **Stateful Inspection** windows.
- Recommendations:
  - Set the gateway's max concurrent connections to 1,000,000.
  - Reduce TCP end session timeout to 5 seconds.

Hardware Configuration for Open Server

Interfaces that are constantly used should not share the same IRQ.

To prevent the use of the same IRQ in such a situation perform the following:

1. Upgrade BIOS to enable IRQ swizzling (for platforms that support this option).

   IRQ swizzling is designed to enable PCI-E based NICs to have different IRQ pools. For example, two PCI-E dual cards are put in two separate busses, they will both receive the same IRQs (16 & 17 for example).

   Upgrading the BIOS to include support for IRQ swizzling will enable the second NIC to have different IRQs (18, 19 for example).

2. Select the ports that do not share the same IRQ for the heavily used network interfaces.

   For example, configure the internal/external/sync subnets so that they will not share IRQ interfaces.

3. Use gigabit based equipment (NICs, switches, cables)

4. Disable hyper-threading

CoreXL Testing

When testing CoreXL it is highly recommended that you simulate a real world scenario and test at least a class C range (that is, approximately a couple of hundred IP addresses). As a result, the connections will be distributed among the different cores and all the cores will be utilized.

Topology

Often during performance tests the simulated topology is such that the device that is being tested lies on the same broadcast domain with many other hosts. The default configuration of SecurePlatform is not customized for such topology since in "real world" topologies the broadcast domain is usually much more limited.

In SecurePlatform the OS ARP cache is limited to 1024, which may result in ARP requests being sent during the tests and as a result, connections will be delayed until ARP learning is complete. This can cause serious performance degradation. It is recommended to increase the ARP cache limit to exceed the number of hosts in the domain by setting `gc_thresh3` on the gateway. For example, execute the following to extend the capacity of the ARP cache to 4096 addresses:

```
echo 4096 > /proc/sys/net/ipv4/neigh/default/gc_thresh3
```