SecureXL and Nokia IPSO™

What Is SecureXL?

SecureXL is the security performance architecture of Check Point VPN-1/FireWall-1 and Nokia security appliances. The architecture offloads many intensive security operations to optimized Nokia IPSO code running on Intel x86 hardware or on network processor hardware. Offloaded security operations include TCP state negotiation, packet forwarding, Network Address Translation, VPN cryptography, anti-spoofing, routing, and accounting. Optimized IPSO code placed at the hardware interrupt level or in a network processor reduces the overhead involved in performing these security operations.

What Does It Do?

SecureXL accelerates firewall and VPN performance – throughput and connection rate – as a result of the reduced overhead. In Nokia security appliances, these improvements result from a software-only upgrade (except for the IP2250 which brings network processor hardware to supplement x86). Nokia security appliances, both new and old, benefit from this new, free software feature (free to customers with current support contracts). SecureXL itself will continue to be enhanced to further improve performance and remove current limitations.

How Does It Do It?

SecureXL accelerates firewall and VPN performance by remembering certain attributes of packets and packet flows that have already been validated by the firewall/VPN application. Thereafter, validation of related packets and connections is delegated to IPSO across the SecureXL API. IPSO either performs this validation natively at the hardware interrupt level on x86 hardware, or supervises execution of further optimized code in attached network processors in the IP2250 security appliance. Both of these approaches involve substantially less computing overhead than required by the firewall/VPN application itself.

The rest of this white paper explains the details of how and why SecureXL accelerates firewall and VPN throughput and connection rate performance, what its requirements and limitations are, and what performance acceleration you can expect to see in your networks.

Firewall Flows and SecureXL

Like Nokia IPSO’s “Firewall Flows” feature, SecureXL reduces the overhead involved in forwarding packets that are parts of flows, which the firewall has already validated. SecureXL replaces, in part, the Firewall Flows feature introduced in earlier versions of IPSO.

Unlike Firewall Flows, SecureXL extends this acceleration to firewall traffic connection rate and to encrypted VPN traffic throughput, as well.

Note that SecureXL makes use of the infrastructure provided by and operates on top of the firewall flows. SecureXL is not mutually exclusive to firewall flows, but actually needs firewall flows mode to be operational in order to be used. Also, IPSO’s slow path is not used with SecureXL.

<table>
<thead>
<tr>
<th></th>
<th>Firewall Flows</th>
<th>SecureXL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerate firewall (unencrypted) traffic throughput</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Accelerate firewall (unencrypted) traffic connection rate*</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Accelerate VPN (encrypted) traffic throughput</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note that connection rate is accelerated by SecureXL only when Network Address Translation (NAT) is disabled. Additionally, VPN (encrypted) packets are not connection rate-accelerated.
SecureXL and Nokia IPSO™

Throughput Acceleration

Packets attempting to establish a new TCP connection (or a comparable UDP connection table entry in the firewall) are handled in the slowpath. Once the first packet is seen by the firewall and suitable connection/flows information is offloaded to IPSO, further packets are handled at IPSO’s interrupt-level code.

The round-trip processing – interrupt driver to application back to driver-level code – is very time consuming relative to the minimal processing necessary for later packets (that can be done entirely at the driver level). Those later packets, determined to be part of an existing, already validated flow, are forwarded directly from the driver level without the overhead of firewall application involvement.

Firewall Flows, and now SecureXL, improve non-encrypted firewall traffic throughput, and now encrypted VPN traffic throughput, by nearly an order-of-magnitude – particularly for small packets flowing in long duration connections.

Following the in-depth description of how SecureXL accelerates connection establishment below, we’ll illustrate the high-level flow of packets through the Nokia SecureXL-accelerated security appliance.

Connection Rate Acceleration

SecureXL provides another form of acceleration. It reduces the overhead in establishing certain kinds of new connections, improving new connection rate (connections per second) and connection setup/teardown rate (sessions per second) as well as throughput in certain high-connection-rate traffic environments.

The principle involved is a simple extension of the Firewall Flows and SecureXL approach to “one-time validation” of a flow. The one-time validation is extended from a particular 5-tuple – source address, destination address, source port, destination port, and protocol (one classic definition of a “flow”, or the definition of a “microflow” by Internet researchers) - to a range, or block, of one or more of these “tuples”. Specifically, “source port” of a flow may be masked off, effectively providing a global match for source port. That is, once a flow is validated and established, a “template” of that flow, with source port masked off creating a global match, is saved and remembered (with a configurable timeout). Any new connection setup that matches four of the 5-tuples, is again handled in the slowpath and not at the driver level. All new connection creation/old connection deletion is handled in the slowpath. However these new connection setup packets matching 4 out of 5 tuples avoid a round trip to the firewall application, thus avoiding the computing overhead. Security is not impacted because IPSO continues to track the state of the new connection using stateful inspection.

Masking the Source Port – Creating a Global Match

The “global match for source port” option is the first stage in the extension of the SecureXL API. Later versions will likely support port and/or address range masking. Below we examine how ports are used in establishing TCP connections to understand how source port masking is particularly effective in reducing connection establishment overhead, and accelerating connection rate and throughput in certain high-connection-rate environments.

A client requesting a connection to a server will initiate the TCP three-way handshake. The client addresses the server typically at a well-known port number depending on the service provided by the server (e.g. port 23 for Telnet, port 80 for HTTP). Together, the server’s IP address and the well-known port number form a “socket address”. The client assigns and pairs an operating system selected port number with the client’s IP address to create a socket address for the reverse direction.

It’s all about the Application Layer Protocol

Now consider higher-level application protocols that involve numerous TCP connections between the client and server – either simultaneously (in parallel) or sequentially, or both. One of these protocols, which accounts for most Internet traffic, is HTTP. SMTP is another example, but we’ll look at HTTP as representative of how SecureXL accelerates connection rate. FTP would not benefit by the source port masking. The FTP control connections are handled by the firewall and the data connections are handled by the firewall until it determines the direction of the data flow.

HTTP and SecureXL–Accelerating Firewall Connection Handling

Web pages consist of multiple HTTP components – text, and perhaps dozens of graphic elements. Using HTTP 1.0, each component is downloaded from server to client using a separate TCP connection. This action involves substantial overhead in connection setup and teardown, and further overhead in protective-firewall connection tracking (firewalls at both ends).
In all cases in the figure above, between a Web Client and a Web Server, TCP connection establishment is initiated by the Web Client, which then sends an HTTP request. The Web Server responds by sending the HTTP component (text or graphic).

**HTTP Request**

Each of the packets from the Web Client that requests an HTTP component from the Web Server has the same source address, destination address, destination port (80), and protocol (HTTP). Only the source port, assigned by the Web Client’s operating system, one per connection, differs in order to create unique socket addresses at the Client for each HTTP request/component (via separate TCP connections for each component).

**HTTP Component**

In going the other direction, each of the packets from the Web Server that build the web page components on the Web Client has the same source address, destination address, source port (80), and protocol (HTTP). Only the destination port differs (it’s been assigned by the Client operating system to that connection).

Let’s look at applying SecureXL at the SERVER Firewall. Once a connection involving a flow to port 80 is approved by the firewall application for the Web Client (resulting from the first HTTP request) a template is created and stored. All subsequent connection setups carrying those additional requests can share that same template “approval” because the fact that the source ports differ is okay. Establishing those subsequent connections does not involve a round trip to the firewall application, and those connections are created much more quickly through the SERVER Firewall as a result.

In the same way, at the CLIENT Firewall, once a connection involving a flow to port 80 is approved by the firewall application (as above), all subsequent connections carrying those additional requests can share that same “approval.” Establishing those subsequent connections does not involve a round trip to the firewall application, and those connections are created much more quickly through the CLIENT Firewall as a result.

SecureXL accelerates subsequent connection establishment through both firewalls, when multiple connections share the same source address, destination address, destination (server) port and protocol.

**HTTP 1.0, 1.1**

HTTP version 1.0 operates as described above, and SecureXL increases the connection establishment rate of the firewall for tracking these connections. This is because HTTP 1.0 creates a separate connection for each HTTP component. The newer HTTP version 1.1 improves the protocol’s performance by permitting not only parallel, but also persistent and pipelined server connections. The server may keep the connection alive after sending the end of a component, which avoids the need to create a new connection to send the next component. HTTP
SecureXL and Nokia IPSO™

1.1 is supported by most web servers and the current generation of browsers as well.

Effect on Other Application-Layer Protocols

The three main connection-oriented application protocols in use today are HTTP, SMTP, and FTP. HTTP behavior was described in detail above.

SMTP is typically transported over TCP. One or more simultaneous connections can be opened to the SMTP server, and they may remain open for the transfer of multiple mail messages. Although the option exists to force a new TCP connection for each mail message, this is not normally done because of the overhead involved. So SMTP is not a connection-intensive protocol.

FTP typically involves long traffic streams and is not connection-intensive.

Other application-layer protocols – RPC, NFS, NNTP, NTP, are not so connection-intensive that they benefit from SecureXL templates.

Current trends are toward the increasing use of streaming protocols to carry audio and video programs, as well as Voice over IP and multimedia conferencing. These streaming protocols are not connection-intensive and benefit from SecureXL templates when streams are unidirectional.

UDP “Pseudo-Connections”

UDP is inherently connectionless. However, FireWall-1 tracks, and Firewall Flows and SecureXL accelerate UDP traffic through the firewall by tracking UDP pseudo-connections. Unlike TCP, applications that use UDP as their transport mechanism have no way of “closing” the pseudo-connections tracked by the firewall – the firewall lets them expire. There are no UDP-based, widely used, traffic-dominating applications that create high “pseudo-connection”-rates – that is, open multiple socket addresses for short durations.

Characterizing Connection Rate using SecureXL with Connection Templates

High connection-rate network environments involve primarily HTTP traffic, particularly HTTP version 1.0. While HTTP 1.1 is significantly less connection-intensive, HTTP likely remains the protocol that generates most new connection requests in enterprise and Internet traffic. At the same time, while overall traffic levels continue to grow, we should expect connection rates to grow less quickly as the migration to HTTP 1.1 unfolds.
Packet Flow through SecureXL

The figure below shows the decision logic for packets flowing through the Nokia SecureXL-accelerated security appliances.

A new packet arrives at the inbound interface. The packet is checked against IPSO’s connection table (which mirrors the VPN-1/FireWall-1 connection table). If there is a 5-tuple match (src, dst, sport, dport, proto), then the new packet is part of an existing flow and is forwarded to the outbound interface for handling (forward, drop, or reject). This path involves the least amount of forwarding overhead and accelerates throughput for packets that are parts of an existing flow.

If the new packet does not match an entry in IPSO’s connection table, then the new packet represents a new flow and requires a new connection table entry. However, if the packet matches an existing connection template, then the new connection table entry can be created (based on information in the connection-template table entry) without a round trip to the VPN-1/FireWall-1 application. The packet is then forwarded to the outbound interface for handling. This path reduces the overhead involved in creating a new connection table entry and accelerates connection rate for new connections that match existing connection templates.

If the new packet does not match an existing connection template, then a round trip to the VPN-1/FireWall-1 application is required to apply the security policy (rules). This path involves the greatest overhead.
The SecureXL Application Programming Interface (API)

The VPN-1/FireWall-1 application communicates with Nokia IPSO software through the SecureXL API. The API supports the exchange of information between VPN-1/FireWall-1 and IPSO relating to packets and packet streams, enabling IPSO to take over validation of subsequent traffic once initially validated by the VPN-1/FireWall-1 application itself.

*The type of information exchanged via the API includes initialization; configuration; flow, security association and connection handling; and statistics. The figure below illustrates the relationship between the VPN-1/FireWall-1 application software and IPSO operating system software running on Intel x86 hardware and (for the IP2250 security appliance only) the network processor hardware.*

The API History

The SecureXL 1.0 API introduced VPN and firewall throughput acceleration. Version 1.5 of the API improved connection rate by further delegating handling of the connection setup packets (except the very first one – the SYN packet) to IPSO.

Version 2.0 increased connection rate by introducing the concept of connection templates. Version 2.1 further incrementally enhances connection rate and security by

- Allowing IPSO to delete connections without getting delete requests from the firewall (auto-expire)
- Allowing IPSO to handle certain short connections completely on its own (delayed notifications)
- Allowing IPSO to implement an enhanced method of tracking the TCP state of TCP connections, including examining the TCP sequence and ACK numbers in addition to the TCP flags

Version 2.12 of the API supports acceleration of multicast traffic.

System Requirements for SecureXL Software

SecureXL requires Nokia IPSO 3.8 (or later) and Check Point VPN-1/FireWall-1 NGAI R55 for IPSO 3.8 (or later).

Hardware

SecureXL is supported on all Nokia IP-series security appliance platforms that run the required software (above). For applications involving large numbers of concurrent connections, additional memory may be advised (compared to IPSO 3.7) because SecureXL consumes more memory per connection than Firewall Flows in earlier versions of IPSO. The increased memory usage reflects the additional information stored in the connection table relative to Firewall Flows. The table below shows the maximum number of concurrent connections for given RAM installations when running SecureXL.

<table>
<thead>
<tr>
<th>ALL IP-Series Security Appliance Platforms except IP2250 RAM Size</th>
<th>Maximum Concurrent Connections</th>
<th>RAM Size</th>
<th>Maximum Concurrent Connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>128 MB</td>
<td>20,000</td>
<td>1 GB</td>
<td>300,000</td>
</tr>
<tr>
<td>256 MB</td>
<td>40,000</td>
<td>2 GB</td>
<td>600,000</td>
</tr>
<tr>
<td>512 MB</td>
<td>150,000</td>
<td>1 GB</td>
<td>400,000</td>
</tr>
<tr>
<td>800,000</td>
<td>2 GB</td>
<td>2 GB</td>
<td>800,000</td>
</tr>
</tbody>
</table>

**CAUTION:** Be aware that configuring the firewall for a higher connection capacity than the installed memory can support may result in erratic operation – disabling SecureXL, dropping packets, or system panic.
VPN Acceleration
Nokia cryptographic accelerators are compatible with SecureXL and may be used as follows:

### Nokia Encrypt Accelerator Cards

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard</td>
<td>IP130, IP350, IP380, IP260, IP265, IP360, IP560</td>
</tr>
<tr>
<td>NIF4107</td>
<td>IP530</td>
</tr>
<tr>
<td>NIF4214</td>
<td>IP380, IP530, IP710, IP740</td>
</tr>
<tr>
<td>NIF4310</td>
<td>IP380, IP530, IP710, IP740</td>
</tr>
<tr>
<td>NIF4400</td>
<td>IP1220, IP1260</td>
</tr>
<tr>
<td>NIF4410</td>
<td>IP1220, IP1260</td>
</tr>
<tr>
<td>NIF4503</td>
<td>IP2250</td>
</tr>
</tbody>
</table>

### Configuring SecureXL

SecureXL is on by default when running IPSO 3.8 and later on the IP2250 security appliance. SecureXL must be enabled for the IP2250 Accelerated Data Path (ADP) technology to function properly, and it should not be disabled. Disabling SecureXL on the IP2250 will severely degrade performance.

On all other platforms, SecureXL is off by default. You may enable it by entering fwaccel on or by using cpconfig.

SecureXL can be controlled and debugged using the other capabilities of the fwaccel command. For full details of the options, enter the fwaccel help command.

**Usage:**
```
fwaccel on | off | ver | stat | cfg <...> | conns | dbg <...> | help
```

**Options:**
- `on` Turns SecureXL acceleration on
- `off [-q]` Turns SecureXL acceleration off
- `ver` Shows SecureXL / firewall version
- `stat` Shows SecureXL acceleration status
- `cfg <options>` Configures SecureXL
- `conns <options>` Prints the SecureXL accelerator’s connection table
- `templates <options>` Prints the SecureXL accelerator’s templates table
- `dbg <options>` Sets debug flags

**Examples:**
- To enable or disable SecureXL, enter `fwaccel on` or `fwaccel off`
  (You may also use `cpconfig`)
- To check overall SecureXL status, statistics, number of accelerated SAs (VPN traffic), etc., enter `fwaccel stat`
- To enable or disable connection templates, enter `fwaccel cfg –p { on | off }`
- To limit the number of connection templates that will be created, enter `fwaccel cfg il nnn`
- To change the default connection template timeout from its default 60 seconds, enter `fwaccel cfg –m xxx`

### What is accelerated by SecureXL?

The following protocols and environments are accelerated by SecureXL, subject to limitations described in the following section.

**Throughput Acceleration**
- TCP traffic
- UDP traffic (unicast)
- IPSec VPN traffic (including support for hardware crypto acceleration)
- Higher layer protocol traffic transported over TCP or UDP (unicast)

**Connection Rate Acceleration**
- Unencrypted TCP traffic when not using NAT
- Unencrypted UDP traffic (unicast) when not using NAT
- Unencrypted higher layer protocol traffic that is transported over TCP or UDP (unicast) when not using NAT
SecureXL 2.1 Features

TCP seq validation by default is disabled in IPSO. Users need to enable this using Voyager to activate this feature. The auto expiry and delayed notification features can also be enabled or disabled using voyager. They are enabled by default. After enabling/disabling any of these features, policy needs to be pushed for the change to take effect.

Connection limits starting from IPSO 3.9:
- Diskless: Disk based
  - 128MB = n/a 22K
  - 256MB = n/a 36K
  - 512MB = 90K 135K
  - 1GB = 225K 360K
  - 2GB = 725K 725K

SecureXL Limitations

SecureXL technology involves the VPN-1/FireWall-1 application delegating certain decisions to software running on the other side of the SecureXL API in IPSO. By making these decisions closer to the network interface, at the hardware interrupt level or in a network processor (IP2250), substantial overhead can be removed (queuing, copying, context switches, etc.). The type of decisions that can be made quickly are the ones that will benefit most from the reduced overhead. On the other hand, complex decisions that take a large amount of time relative to the overhead involved don’t have as much opportunity for optimization.

The SecureXL API and IPSO support delegation of those decisions that benefit most from reduced overhead – involving the most common type of traffic and security policies. The evolution of SecureXL over time will address the most significant remaining limitations.

Certain types of traffic, and certain elements of the security policy, can negate the benefits of SecureXL, either for particular packets or for all traffic. For example, when setting up a new connection, in order to execute a rule where the source or destination is a domain, this decision (which is not delegated through the SecureXL API to IPSO) must be executed by the VPN-1/FireWall-1 application itself. All new connections would need to be examined with this rule by the application itself rather than by IPSO, and so the use of this rule negates SecureXL’s template connection rate acceleration for all traffic matching this rule and below.

To get the most out of SecureXL performance acceleration, the choice of Check Point applications and features, the rules that make up the security policy, and even potentially the ordering of the rules, should be carefully chosen based on the information below.

Incompatible Check Point Applications

- FloodGate-1 (automatically disables SecureXL and enables Firewall Flows)
- SmartView Monitor (traffic charts and counters don’t account for packets that the SecureXL device handles)

Incompatible Check Point Features

- Prior to IPSO-3.8.1 Smart Defense TCP Sequence Verifier (sequence verification not enforced when SecureXL is enabled even though management console indicates that it is enforced). IPSO-3.8.1 onwards has support for Sequence verification in IPSO’s SecureXL code. This feature is disabled by default, but can be enabled using voyager, as mentioned above.
- Smart Defense Fingerprint Scrambling (traffic will not be accelerated)

Firewall Rule Limitations

Certain security policy rules and rule properties invoke extensive algorithms that are not replicated across the SecureXL API. SecureXL wouldn’t necessarily enable significant acceleration even if they were replicated because of their complexity relative to application overhead. For optimum performance, the security policy should be designed, where possible, avoiding these rules and rule properties.

The following rule properties present in the security policy will disable connection-rate acceleration for all traffic. (Throughput acceleration is not inhibited by the presence of rules with these properties.)

- Service with a port number range
- Service type “other” with a match expression
- Service type RPC, DCOM, or DCE-RPC
- Service with “enable reply from any port” checked
- Source or destination is a domain or a dynamic object
- Time object associated with the rule
- Client or session authentication involved with the rule
- SYN Defender (the entire 3-way handshake must be supervised by the FireWall-1 application, slightly reducing the effect of connection-rate acceleration – most significant performance impact on short duration connections)
The following rule properties present in the security policy will disable throughput and connection-rate acceleration for all traffic.

- Rules with action “encrypt” on an interface that does not support cryptography
- Rules where the source or destination of the rule is the gateway itself
- Rules where the service has an INSPECT handler (e.g. FTP control connection)
- Rules with Security Servers or services with resources
- Rules with user authentication
- Rules for non-TCP/UDP/GRE/ESP connections

Traffic Limitations

The following traffic is not throughput or connection-rate accelerated by SecureXL.

- Multicast traffic
- Directed broadcast traffic
- Traffic across an Access Control List-enabled interface
- Traffic whose Protocol field in the IP header is not TCP or UDP (e.g. ICMP, IGRP, etc.)
- IPv6 traffic
- VPN encryption algorithms that are not supported by the hardware.
- IP compression enabled for VPN traffic.

The following traffic is not connection-rate accelerated by SecureXL.

- VPN
- Complex connections such as FTP, H.323...
- Non-TCP/UDP connections

Environment Limitations

The following environment disables connection rate acceleration for the traffic that the environment is applied to by SecureXL.

NAT

FTP

VPN traffic
Disclaimer: Use of the word secure is intended to describe the functionality of the product or feature described, and is not intended to extend a warranty to the purchaser or to any end user that the product or feature described is completely secure and invulnerable to random attacks.

Nokia is the world leader in mobile communications, driving the growth and sustainability of the broader mobility industry. Nokia is dedicated to enhancing people’s lives and productivity by providing easy-to-use and secure products like mobile phones, and solutions for imaging, games, media, mobile network operators, and businesses. Nokia is a broadly held company with listings on five major exchanges.

For more information, please visit http://www.nokia.com/forbusiness.