The PA-5000 Series Architecture

The Evolution of the Single Pass Parallel Processing Architecture

October 2011

Palo Alto Networks
3300 Olcott St
Santa Clara, CA
95054
www.paloaltonetworks.com

Table of Contents

Executive Summary: The Need for a Single Pass Architecture

For many years, the goal of integrating threat prevention services into the firewall has been pursued as a means of alleviating device additional security devices for functions like IPS, network antivirus, and more. The pursuit of integrating threat prevention functions into the firewall makes perfect sense – the firewall is the cornerstone of the security infrastructure.

Current integration iterations carry a variety of different labels – deep inspection, unified threat management (UTM), deep packet inspection, and others. What each of these iterations share is a common failure which is a lack of consistent and predictable performance with security services enabled. Specifically, the firewall functions are capable of performing at high throughput and low latency but when the added security functions are enabled, performance decreased while latency increased.

The Palo Alto Networks single pass parallel processing architecture addresses the integration and performance challenges with a unique, single pass approach to packet processing that is tightly integrated with a purpose-built hardware platform.

- **Single pass software:** By performing operations once per packet, the single pass software eliminates many redundant functions that plague previous integration attempts. As a packet is processed, networking is performed once, policy lookup is performed once, application identification and decoding is performed once, and signature matching for any and all threats and content is performed once. This significantly reduces the amount of processing overhead required to perform multiple functions in one security device. The single pass software uses a stream-based, uniform signature matching engine for content inspection. Instead of using separate engines and signature sets (requiring multi-pass scanning) and instead of using file proxies (requiring file download prior to scanning), the single pass architecture scans traffic for all signatures once and in a stream-based fashion to avoid latency introduction.

- **Parallel processing hardware:** The single pass software is then integrated with a purpose-built platform that uses dedicated processors and memory for the four key areas of networking, security, content scanning and management. The computing power within each platform has been specifically chosen to perform the processing intensive task of full stack inspection at multi-Gbps throughput levels.

The resulting combination delivers the horsepower required to achieve consistent and predictable performance at up to 20 Gbps of throughput making the goal of integrated firewall and threat prevention a reality.

Key Benefits of Integrated Security

It is important to point out that integrating key security functions into the firewall makes perfect sense, or put another way, this is not integration for the sake of integration. Integration will bring many benefits to any organizations, and they are important to consider when discussing the single pass approach taken by Palo Alto Networks.

- **Network complexity:** Over the last several years, every new security need resulted in a new security device to solve it. As the number of security requirements increased, the number of devices deployed at key network junction points has increased to an unmanageable point. There's no longer enough data ports, port mirrors, network taps, rack space, or power to accept additional devices into the network. Integration (if done well) starts to simplify the network.

- **Network performance:** With every new device comes additional latency, throughput chokepoints, routing issues, and more. Integration (if done well) can reduce network latency and the number of chokepoints traffic must pass through.

- **Functional holes:** There are several basic pieces of information useful for setting security policy – irrespective of the function. Source user or IP address, application, application function, URL category, port, protocol, and traffic destination. But each device acquires this information in unique ways, or in many cases, isn't capable of acquiring some of the pieces. These gaps and inconsistencies significantly impact the security effectiveness. Integration (if done well) allows the information to be collected once and applied in a single security policy.

- **Operational management:** Managing the complexity of loosely interconnected set of devices is not a simple task. Separate management systems, functional holes, unknown functional overlaps, and network complexity all contribute to costs and potentially ineffective network security. Integration (if done well) simplifies security management through fewer consoles, fewer functional gaps and for effective security coverage.

- **Total cost of ownership:** The cost of purchasing separate devices for each security functional requirement, maintaining the equipment, and operational costs all add significantly to the total cost of ownership. Integration (if done well) can significantly reduce each of these costs.

These are just a few of the more significant integration benefits – assuming that it is done well. If the benefits are so significant, the obvious question becomes, why have the previous attempts failed?

Problems With Traditional Approaches to Integration

The tradition approach to integrating security functions together is largely flawed for two reasons:

- **Flawed traffic classification:** The traditional approach to security integration is to add functions on top of a foundational firewall. This firewall classifies traffic by protocol and port number (e.g. TCP/80), which is essentially meaningless for today's applications which use non-standard, non-unique, and/or dynamically selected ports. All further security functionality is then based on a flawed initial traffic classification. This topic is covered further in other information from Palo Alto Networks.

- **Flawed integration methodology:** Previous attempts to integrate security functionality is based on simply collapsing multiple functions into one operating system and chassis. This isn't integration, it is consolidation, and the difference is critical. Consolidation simply takes multiple products and stuffs them into a single device. In many cases, management and hardware is still separate, but there is an illusion of integration because the functionality is performed in one device. In other cases, the functions all run on the same general purpose CPU, draining system resources with each addition function that is activated.

The benefits of integration cannot be achieved with these glaring issues with previous attempts.

Palo Alto Networks Single Pass Software Architecture

While a seemingly trivial and obvious approach, security software that looks at traffic in a single pass is unique to the Palo Alto Networks next-generation firewall. This approach to processing traffic ensures that each particular task is performed only once on a set of traffic. Key processing tasks are as follows:

- **Networking and management functionality:** at the foundation of all traffic processing is a common networking foundation with a common management structure.
- **User-ID:** maps IP addresses to active directory users and users to groups (roles) to enable visibility and policy enforcement by user and group.
- **App-ID™ (Application identification):** a combination of application signatures, protocol detection and decryption, protocol decoding, and heuristics to identify applications. This application identification is carried through to the Content-ID functionality to scan and inspect applications appropriate to their use as well as to the policy engine.
- **Content-ID:** a single hardware-accelerated signature matching engine that uses a uniform signature format to scan traffic for data (credit card numbers, social security numbers, and custom patterns) and threats (vulnerability exploits – IPS, viruses, and spyware) plus a URL categorization engine to perform URL filtering.
- **Policy engine:** based on the networking, management, User-ID, App-ID, and Content-ID information, the policy engine is able to use a enforce a single security policy to traffic.

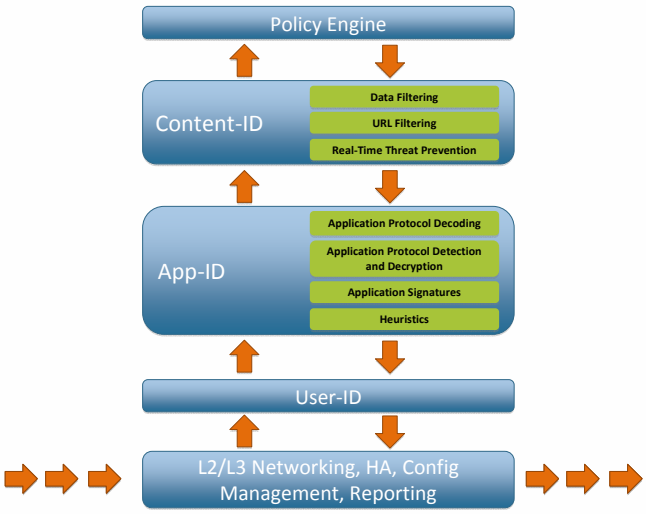

Image 1: Traffic flow for the single pass software architecture.

Scan it all, scan it once

One of the key elements to the single pass architecture is summed up accurately and succinctly with the phrase "scan it all, scan it once".

- **Common protocol decoding engine:** The first key component to the single pass architecture is the use of a common protocol decoding engine that is used for all traffic. The decoding engine is used to pick apart an application stream to determine what the different pieces are – for example, where does a file transfer start and stop, what is the file type, when is the user posting data versus downloading data, when is a command being executed. All of this information is then used as the basis for scanning the content for files, data, threats, and URLs. By performing the content scanning task once instead of multiple times, significant processing power is saved as this is one of the most processing-intensive tasks for a security device to perform.

- **Stream-based signature engine:** The use of a stream-based engine replaces several components commonly used in other solutions - a file proxy for data, virus, and spyware, a signature engine for vulnerability exploits, and an http decoder for URL filtering. By using one common engine, two key benefits are realized. First, unlike file proxies that need to download the entire file before they can scan the traffic, a stream-based engine scans traffic real time, only reassembling packets as needed and only in very small amounts. Second, unlike traditional approaches, all traffic can be scanned with a single engine, instead of multiple scanning engines.

Advantages/Disadvantages of Stream-Based Engine

One detail that shouldn't go without discussion is the advantages and disadvantages of a stream-based scanning engine versus a file proxy engine. The benefits of a stream-based engine are straightforward:

Scalability: the stream-based engine requires significantly less memory and processing power since it doesn't need to store the entire file while its downloading prior to scanning. Think of 5,000 users simultaneously downloading 5,000 different files and a file proxy trying to manage all of them – it just doesn't work. A stream-based engine scans the downloads as they pass through, a much more feasible approach for scanning large amounts of data.

- **Low latency:** the stream-based engine process and forwards the file as it receives it, scanning it with sub millisecond latency unnoticed by the end user. File proxies on the other hand can introduce latency into the 10's of seconds – a recent Network World test of a UTM device showed 45 second latency on downloads.

- **Common processing:** using a stream-based engine enables one processing engine for all traffic whereas a file proxy can't scan for vulnerabilities and must therefore be part of a multi-pass approach.

On the other hand, there are several key trade-offs with the stream-based engine that should be considered:

- SMTP, POP3, IMAP: stream-based engines work very well for most applications, but not for blocking viruses, spyware, or data over traditional email applications like SMTP. While alerting works well, without actually proxying the connection, blocking attachments within an email message will often just cause a continuous retransmission of the attachment over SMTP. In addition, it isn't possible to quarantine the email message. Usually this isn't a problem as the email server is already surrounded by one or more layers of antivirus.

- The number of compressed formats that can be scanned is limited to zip and gzip (without password encryption) as these are the only two compression formats that compress in blocks of data instead of the entire file as one compressed block. This is typically not a problem as these are the most common compression algorithms and this is supplemented with file type scanning and alerting so that other file types can be monitored and potentially blocked from traversing certain network segments or certain applications.

Keeping the goal of integration and performance in mind, Palo Alto Networks chose to implement a stream-based scanning engine based on the fact that the advantages outweighed the disadvantages.

Hardware Acceleration

One conventional belief that is now rendered obsolete is the notion that while firewalls can be hardware accelerated, application layer scanning for content cannot. The main challenge with accelerating scanning in hardware was due to the architectural approach described above – proxying files and multiple scanning engines are not conducive to hardware acceleration. The second challenge to accelerating content scanning in hardware was that it was often viewed as an afterthought and was not architected into the hardware and software from the outset. With Palo Alto Networks single pass parallel processing architecture, hardware acceleration is provided for each of the major functionality blocks:

- Networking: per packet routing, flow lookup, stats counting, NAT, and similar functions are performed on dedicated network processor.
- User-ID, App-ID, and policy engine all occur on a multicore (up to 16 cores) security processor with hardware acceleration for encryption, decryption, and decompression.
- Content-ID performs the signature lookup via a dedicated FPGA with dedicated memory.
- Management functionality is provided via a dedicated control plane processor that drives the configuration management, logging, and reporting without touching data processing hardware.

The architecture defined above was initially brought to market in 2007 with the release of the PA-4000 Series. As a proof point to the scalability of the architecture, subsequent new hardware platforms (PA-2000 Series, PA-500) used the exact same architecture, albeit with different processors appropriate for the performance goals.

PA-5000 Series Hardware Architecture

The previous section introduced the four key elements of the Palo Alto Networks Next Generation hardware architecture:

- Control Plane Processor
- Network Processor
- Multi-Core Security Processor
- Signature Match Engine

The PA-5000 Series effectively enhances these key elements to deliver double the performance so that the next-generation firewall features could be further extended into the datacenter and service provider markets.

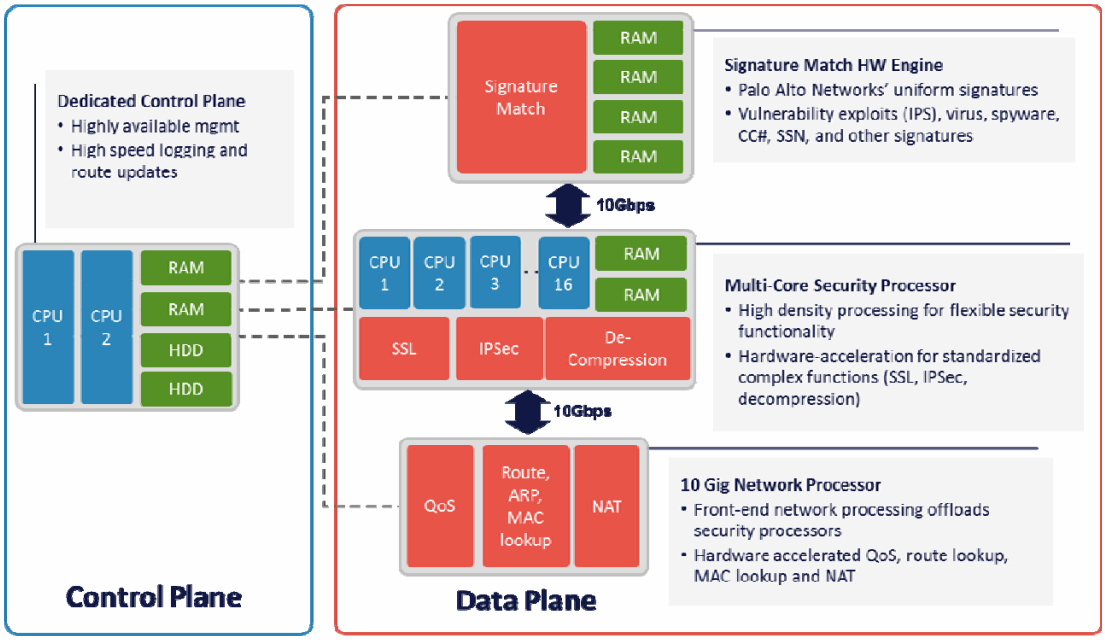

Image 2: PA-4000 Series hardware architecture.

Doubling throughput performance is not as simple as doubling the performance of the key processor components. Given today's technologies, that simplistic mechanism to achieve performance increases doesn't always work. In some areas of the architecture, this methodology will work, for example, on the control plane (utilizing a larger CPU with more memory) and on the network processor.

However, for the remaining architectural elements, the multi-core security processor and the signature match engines, this simple methodology of doubling the performance by doubling processor speed was not possible, given today's FPGAs (signature match engine) and multi-core processors.

Given these limits, Palo Alto Networks made some slight modifications to the architecture of the PA-5000 Series to achieve the performance desired, using existing technology, and doing so in a way that fundamentally kept the architecture of this system virtually identical to that of the previous generations. The architectural diagram below displays the new PA-5000 Series hardware architecture.

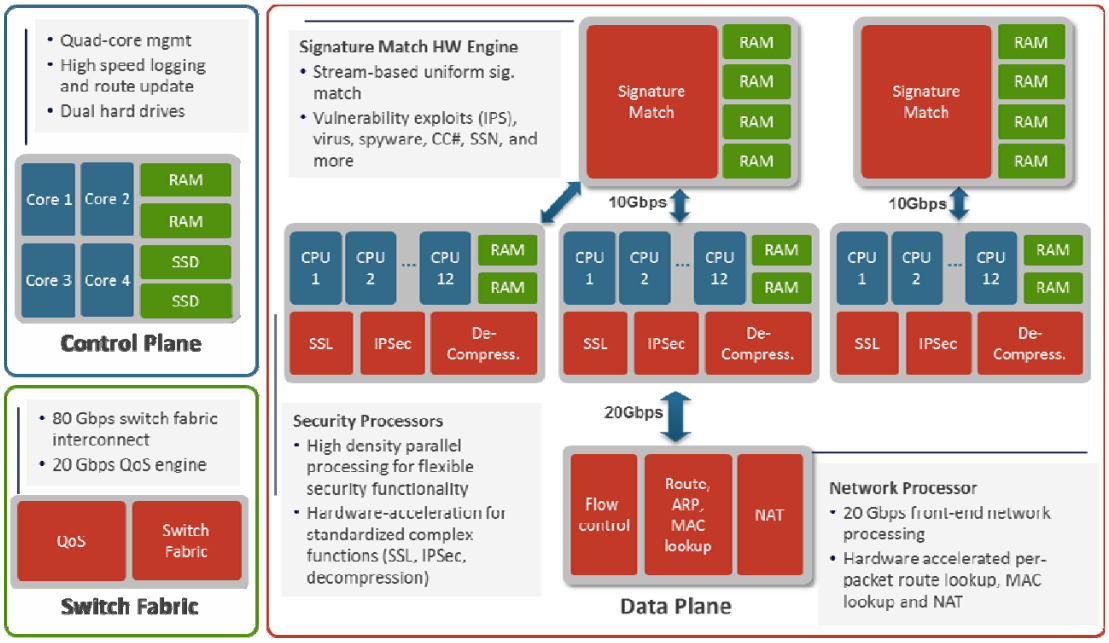

Image 3: PA-5000 Series hardware architecture.

The architecture diagram shows that the control plane processor has been increased in both performance and capacity, using a quad core Intel CPU with 4GB of DRAM, and dual solid state storage drives (SSD) for high performance and reliability. Similarly, the network processor is still a single element, but now supports 20Gbps bandwidth.

The most significant change in the architecture, however, is in the multi-core CPU area where the computationally challenging security processing is executed. As a single, massive CPU was not available that met the data plane CPU performance goals of the PA-5000 Series, the decision was made to continue using multiple multi-core CPUs to satisfy the goal of doubling performance. Each of the three multi-core CPUs in the PA-5000 Series provides approximately the same level of performance as the single data plane CPU in the PA-4000 Series, but there are now three of them, effectively tripling the overall performance.

The decision to 'go wide', using multiple multi-core CPUs presented some additional challenges not seen in the previous platform; how does the system distribute the network load across these CPUs? The solution to this problem was to designate one of the CPUs (in this case, the 'one on the left') as the 'master CPU', responsible for distributing the new TCP or UDP sessions across all of the CPUs in the system for further processing. The master CPU is also responsible for handling a few special types of traffic, such as IPSec VPN and 'non-TCP/UDP traffic'. As this master CPU has these other tasks to perform, it is only tasked with a small percentage of the 'regular' traffic traversing the firewall (a proprietary, dynamic algorithm is used for load distribution between the CPUs for network traffic), with the bulk of the traffic processing handled by the other 2 multi-core CPUs.

Similarly, FPGA technology did not exist to double the performance of the signature match engines, so the same approach was used to increase the performance of this aspect of the architecture. Instead of using a single signature processor, as was done on the PA-4000 Series, the PA-5000 Series uses two FPGAs, each with high bandwidth access to all three of the multi-core security processors.

Single Pass vs Multi-Pass Architecture Comparison

The initial comparison to providing multiple security functions in discrete devices is very obvious – each one of the described blocks in the single pass architecture will be performed by each device (assuming they can perform all of the functions). The duplication of processing is staggering in this case. Additionally, existing attempts to integrate security functions into a single device are often merely sheet metal integration, where often the networking and management functions are integrated, but elements of traffic classification, protocol decoding, file proxying, and signature matching are performed with separate software and sometimes separate hardware. The diagram below shows a worst case view at discrete devices multi-pass approach:

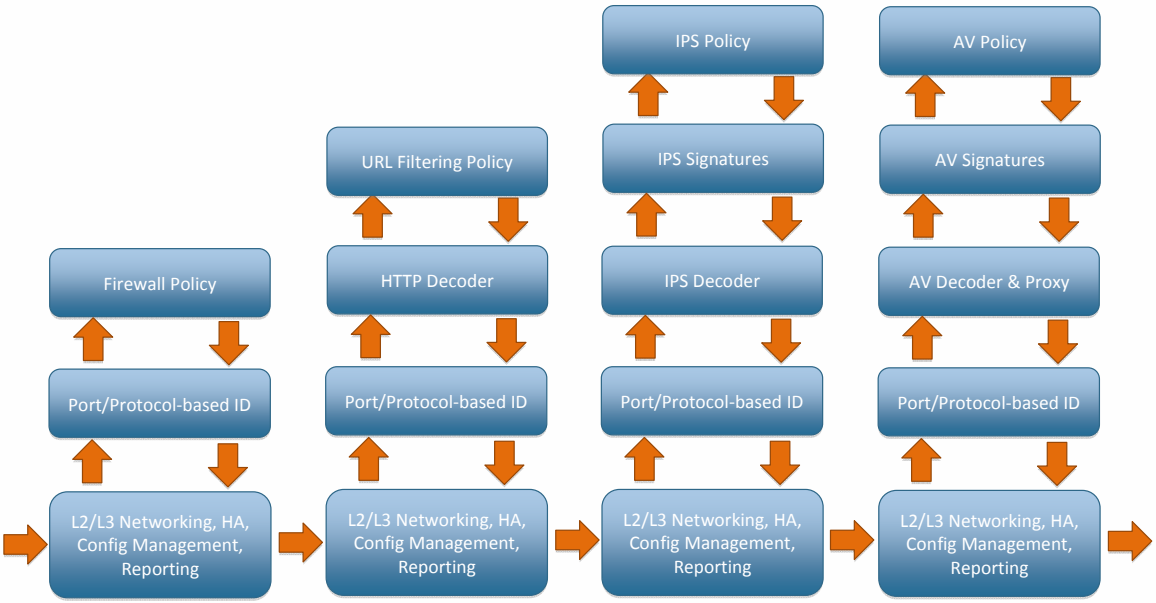

Image 4: Traffic flow for multi-pass hardware architecture.

The above diagram assumes that there are discrete devices performing each function, which results in multiple passes through the networking layer, traffic classification, decoders, signature engines, and policy tables. Each one of these passes generates processing overhead, latency introduction, throughput degradation, and operational costs to keep it all functioning. Some basic savings has been achieved in that the networking layer and port/protocol identification are often collapsed into a single pass. However, most of the heavy lifting – file proxies, application decoding, signature engines, policy enforcement are often still separate multi-pass functions with all the overhead and shared processing.

Single Pass Parallel Processing Architecture Throughput and Latency Results

To put some specifics around what the single pass architecture is able to provide in terms of performance, the following set of data describes the performance of the PA-5060 with all services turned on. These tests are designed to fully stress the device. Most networks would have a mixture of traffic and would see a blending of the performance shown here.

	Firewall: Basic UDP RFC 2544 Test	**Firewall Throughput with App-ID**
PA-5060	20 Gpbs/17µs latency (1,518-byte packets) 7.8 Gbps/11µs latency (64-byte packets)	16.8 Gbps
Test Description	All UDP traffic with specific packet sizes	All HTTP traffic with 512KB response per session
	Firewall w/ App-ID and Threat Prevention (DSRI*)	Firewall w/ App-ID and Threat Prevention
PA-5060	16.8 Gbps	8 Gbps
Test Description	All HTTP traffic with 512KB response per session	All HTTP traffic with 512KB response per session

**DSRI = Disable Server Response Inspection, typical in datacenter deployments.*

A few key performance metrics are listed in the table above. Important takeaways from the above information include:

- Firewall throughput levels are based on traditional firewall testing methodology but with full App-ID traffic classification. The Palo Alto Networks next-generation firewall is designed to scan all traffic for all applications.
- Threat prevention throughput, while scanning for all threats, including vulnerability exploits, viruses, spyware, and sensitive data.

Conclusion

Back to the original question: why is integrated security and a single pass architecture needed? As the number of needed security functions continues to increase, there are two options: add another device or add a function to an existing security device. With a single pass architecture, Palo Alto Networks has made it possible to add a function to a next-generation firewall instead of adding another security device, and in such a way that the integrated approach actually offers benefits and advantages that discrete devices cannot accomplish. There will still be a need in specific cases for discrete devices where highly specialized functionality is required, but for the majority of cases, integrated security is becoming a viable option.