NEXT-GENERATION SECURITY PLATFORM

# Palo Alto Networks Single-Pass Architecture: Integrated, Prevention-Oriented Security

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

Current integration approaches carry a variety of labels – deep inspection, unified threat management (UTM), deep packet inspection, and others. These approaches share a common problem, which is a lack of consistent and predictable performance when security services are enabled. Specifically, the base firewall functions are capable of performing at high throughput and low latency, but when the added security functions are enabled, performance decreases while latency increases.

More importantly, these traditional approaches to integration limit security capability. This is because a "sequence of functions" approach is inherently less flexible than one in which all functions share information and enforcement mechanisms.

The Palo Alto Networks *Single-Pass Architecture* addresses these performance and flexibility challenges with a unique single-pass approach to packet processing.

- **Performance:** By performing operations once per packet, the single-pass architecture eliminates many redundant functions that plague previous integration attempts. As packets are processed, networking, policy lookup, application and decoding, and signature matching for any and all threats and content are performed only once. This significantly reduces the amount of processing overhead required to perform multiple functions in one security device. For content inspection and threat prevention, the single-pass architecture uses a stream-based, uniform signature matching engine. Instead of using separate engines and signature sets (requiring multiple passes), and instead of using proxies (requiring download prior to scanning), the single-pass architecture scans traffic for all signatures once – avoiding the introduction of latency.
- **Flexibility:** The single-pass architecture also supports superior security posture relative to traditional integration attempts. This is because the architecture performs full-stack inspection up-front, and then makes all resulting context available to all security enforcement options (including threat prevention). This stands in contrast to traditional integration

approaches in which full context is not shared between all enforcement options.

Implemented in a variety of form factors (both physical and virtual), our next-generation firewalls based on Single-Pass Architecture are the high-performance foundation of a security platform that stops modern threats.

### 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 integration's sake. Integration will bring many benefits to any organization, and they are important to consider when discussing the single-pass approach taken by Palo Alto Networks.

- **Network complexity:** Traditionally, 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 increased to an unmanageable point. There are no longer enough data ports, port mirrors, network taps, rack space, or power to easily accept additional devices into the network. Integration – if done well – starts to simplify the network.
- **Network performance:** With every new device, additional latency, throughput chokepoints, routing issues, and more are introduced. 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 that are useful for setting security policy, irrespective of the function. These include: source user or IP address, application, application function, URL category, port, protocol, and traffic destination. But each device or scanning process acquires this information in unique ways, or in many cases, is not capable of acquiring some of the pieces. These gaps and inconsistencies significantly impact security effectiveness. Integration – if done well – allows the information to be collected once and applied in a single, flexible set of security policies.
- **Operational management:** Managing the complexity of a 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 and functional gaps, and provides more 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 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 traditional approach to integrating security functions 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 type of firewall classifies traffic by protocol and port number (e.g., TCP/80), which is essentially meaningless for today's applicaitons which often use non-standard, non-unique, and/or dynamically selected ports. All further security functionality is then based on this flawed initial traffic classification. This topic is covered further in other articles from Palo Alto Networks.
- **Flawed integration methodology:** Previous attempts to integrate security functionality are 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 additional function that is activated.

The benefits of integration cannot be achieved without addressing these glaring issues.

### Palo Alto Networks Single-Pass 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.

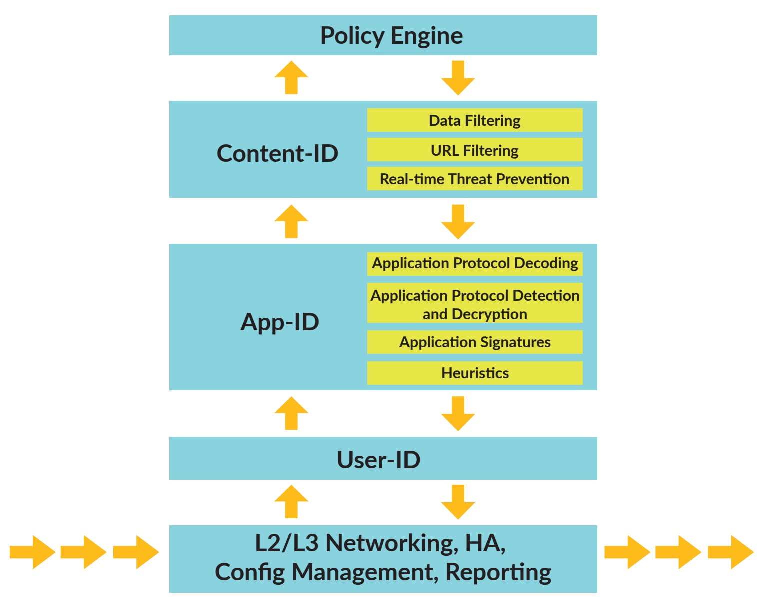


**Figure 1:** Single-Pass Architecture Traffic Flow

- **User-ID™:** Maps IP addresses to (e.g., Active Directory) users and users to groups (roles) to enable visibility and policy enforcement by user and group.
- **App-ID™:** 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™:** Single hardware-accelerated signature matching engine that uses a uniform signature format to scan traffic for data (e.g., credit card numbers, Social Security numbers, and custom patterns) and threats (e.g., 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 and enforce a single security policy to matching traffic.

### *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:** A 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, and 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 in 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 a Stream-Based Engine*

One detail that should not 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 it's 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 file downloads as they pass through, which is a much more feasible approach to scanning large amounts of data.
- **Low latency:** The stream-based engine processes and forwards the file as it receives it, scanning it with submillisecond latency unnoticed by the end user. File proxies, on the other hand, can introduce latency into the 10s of seconds.
- **Common processing:** Using a stream-based engine enables one processing engine for all traffic; whereas a file proxy cannot scan for vulnerabilities and must therefore be part of a multi-pass approach.

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 protocols, such as SMTP. While alerting works well, without actually proxying the connection, such blocking attachments within an email message will often cause a continuous retransmission of the attachment over SMTP. In addition, it is not possible to quarantine the email message. Usually, this is not 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 applications.

Keeping the goal of integration and performance in mind, Palo Alto Networks chose to implement a stream-based scanning engine.

### *Hardware Acceleration*

Implementations of Palo Alto Networks single-pass architecture exist in both virtual and physical form factors. For physical appliances, the single-pass architecture is accelerated by a purpose-built hardware architecture. That hardware architecture is outlined briefly in this section.

One conventional belief that has been 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 traditional architectural approach described earlier – 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 our single-pass architecture, we provide hardware acceleration for each of the major functionality blocks, as illustrated in the example of the PA-7080 architecture shown on the next page in figure 2:

- Network processing is based on per-packet routing, flow lookup, stats counting, NAT, and similar functions and is performed on dedicated network processors.
- User-ID, App-ID, and policy enforcement. This occurs on multicore security processors with hardware acceleration for encryption, decryption, and decompression.
- Signature Matching for Content-ID performs the signature lookup via dedicated FPGAs 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.

### *Single-Pass vs. Multi-Pass Architecture Comparison*

The initial comparison to providing multiple security functions in discrete devices is 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 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 as well. Figure 3 below shows a worst-case view of discrete devices with a multi-pass approach:

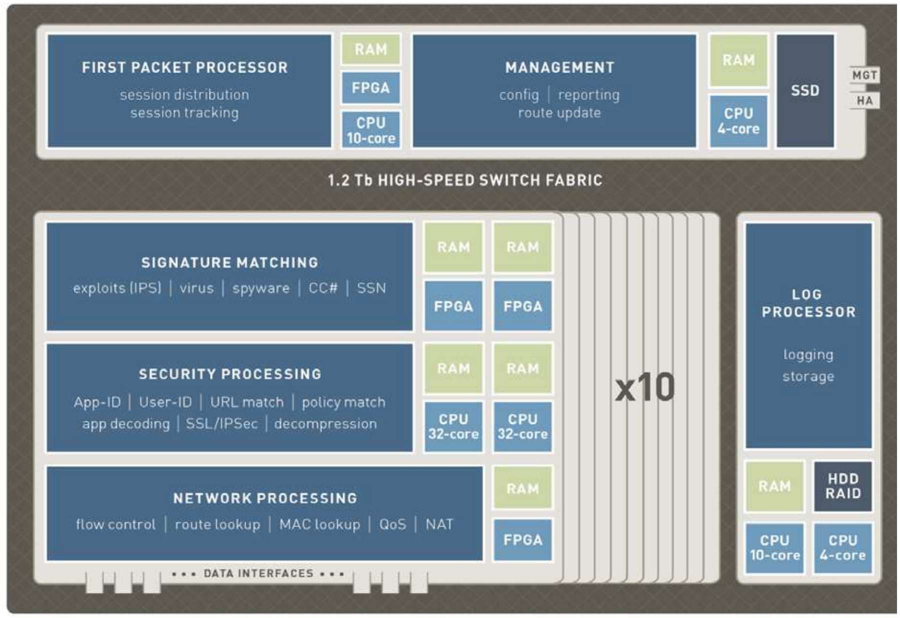


**Figure 2:** PA-7080 Hardware Architecture

The figure 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 cost saving 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, including file proxies, application decoding, signature engines, and policy enforcement are often still separate functions with overhead that competes for shared processing.

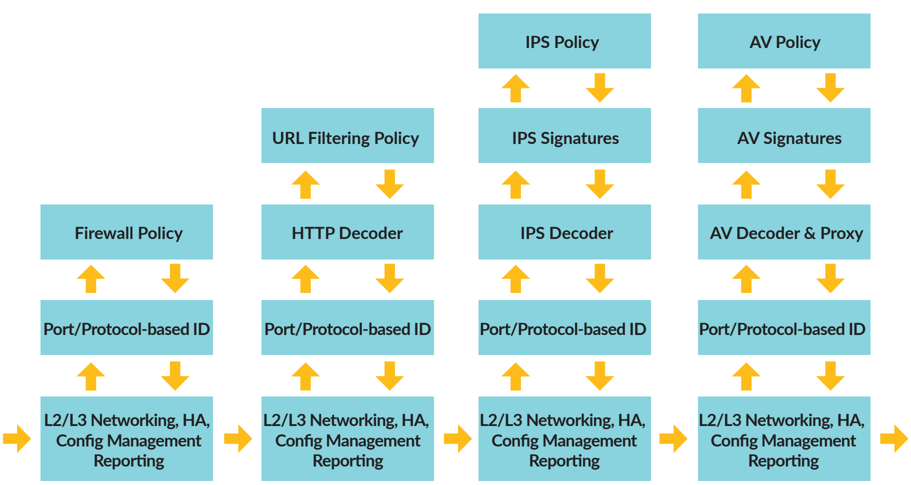


**Figure 3:** Traffic flow for multi-pass architecture

### Conclusion

Back to the original question: why are integrated security and a single-pass architecture needed? As the number of needed security functions continues to increase, there are two options: add another security device or add a function to an existing device. With the 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. There will still be a need for discrete devices in specific cases where highly specialized functionality is required; but for the majority of cases, integrated security is now a viable option.

4401 Great America Parkway
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com


PAN_WP_SPA_092815