

IPS vs. IDS

Similar on the surface, polar opposites underneath

Business white paper





Table of contents

Introduction	3
Stability: Network outage vs. security analyst annoyance	3
Deterministic network performance: Peak vs. average throughput design... ..	3
False positives: Wasted time vs. blocking legitimate traffic	4
Reduce false negatives: Network performance vs. decoding fidelity	5
IDS: Theory vs. reality	6
Conclusion	7

Introduction

A common notion is that an Intrusion Prevention System (IPS) is nothing more than an Intrusion Detection System (IDS) deployed in line with blocking capabilities. This paper explains why that notion is incorrect.

Although IPS and IDS both examine traffic looking for attacks, there are significant differences. IPS and IDS both detect malicious or unwanted traffic. They both do so as accurately as possible, depending on the speed of the network. But an IPS is an inline device designed for automatic enforcement of network policy, whereas an IDS is an out-of-band device designed as a forensic tool for security analysts.

This difference in deployment and utility changes the:

- Emphasis on device design requirements
- Methods that hackers use to attack the devices

Not surprisingly, these changes lead to different engineering designs and technology that may be ideal for IDS but may be suboptimal for IPS, or vice versa.

IPS and IDS share four basic requirements:

- Stability
- Deterministic network performance
- Reduce false negatives
- Reduce false positives

Although these requirements appear to be similar, the differences between IPS and IDS deployment and purpose cause substantial distinctions in prioritizing the requirement, the meaning of the requirement, and implementation options available for meeting the requirement.

Stability: Network outage vs. security analyst annoyance

Although both an IDS and IPS should be stable, the effect of an IPS crash is dramatically different from an IDS crash. If an IDS crashes, it is annoying to the security engineer and causes a temporary security blind spot until the device reboots. If an IPS crashes, the network may go down. Consequently, product stability takes on a much higher priority for an IPS than for an IDS.

Deterministic network performance: Peak vs. average throughput design

Network performance has two dimensions: throughput and latency. Throughput is the processing capacity of

the device, measured in packets per second (pps) or bits per seconds (bps). Latency is the amount of time between when a packet is received and when it is processed and released.

For out-of-band IDS, the processing capacity must at least match the average network load. Latency between capture and reporting can range from a few seconds to a few minutes. To meet these requirements, out-of-band devices like IDSs can absorb temporary bursts in traffic by storing packets in large memory buffers. A buffering implementation reduces the strain on the IDS. With no buffering, the IDS would have to match the network's peak load, which is typically five to 10 times higher than the average load.

IDS being a logging device, provides data to humans—the time lag between receiving a packet and logging an alert can be a few seconds without causing any harm. Indeed, in many environments, delays up to several minutes are acceptable. With these latency requirements, large buffers can be used.

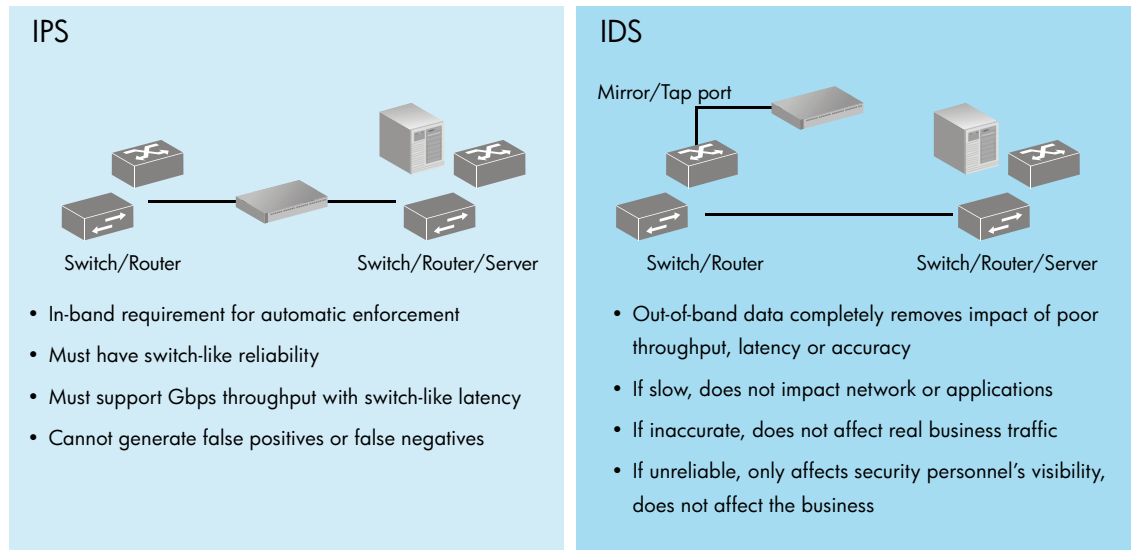
For an in-line IPS, the processing capacity must match peak network load and the latency must be at par with that of the fastest connection on that network. In the network core, this typically means one or more Gbps of processing capacity and latency under 200 microseconds. At the network edge, the throughput requirement drops to a few hundred Mbps and latency can be on the order of one to 10 milliseconds. If IPS throughput is less than peak load, the IPS will be a bottleneck in the network. If the IPS adds significant latency, application response time and TCP throughput will suffer.¹

The use of large memory buffers to absorb traffic bursts is wholly unacceptable for an IPS, as such buffering would significantly increase latency. For example, consider a network link with 100 Mbps average load that bursts to 1 Gbps for one second. An IDS could meet these requirements with 125 megabytes of buffering (1 Gbps * 1 second / 8 bits-per-byte) and a processing capacity of 100 Mbps. If the same IDS were deployed inline, the latency during bursts would rise to nine to 10 seconds, effectively creating an intermittent network outage.² Compare this to a power plant that has to burn more fuel during peak hours to generate more megawatts of power. If a power plant was designed for average demand, there would be blackouts during peak hours.

For an inline IPS, the processing capacity must match peak network load and the latency must be at par with that of the fastest connection on that network.

¹ TCP throughput is governed in practice by RTT, a measure of network latency between two TCP endpoints.

² Moving from a buffering design to an in-line design is a major rewrite of the core of the IDS. Most legacy IDS vendors simply relabeled their products as IPSs and still exhibit such design flaws today (Note: the largest value HP TippingPoint has observed in their lab is 167 seconds of latency for an ICMP echo request packet). Such latencies are functionally equivalent to a network outage.



An IPS is designed to mitigate attacks in real time.

An IDS is designed to alert a security analyst of suspicious behavior.

False positives: Wasted time vs. blocking legitimate traffic

All IPS and IDS vendors strive to reduce false positives. But the meaning and importance of “false positive” is different for IPS and IDS. The difference stems from the design goals of IDS and IPS. An IDS is designed to alert a security analyst of suspicious behavior. An IPS is designed to mitigate attacks in real time.

An IDS false positive is an alert that did not result in an intrusion. It may be that the system under attack was not vulnerable to the attack, or that the detection mechanism may be faulty, or that the IDS detected an anomaly that turned out to be benign. An IDS false positive causes a security analyst to expend unnecessary effort.

When an IPS has a false positive, the primary concern is that legitimate traffic will be blocked. Most organizations consider blocking legitimate traffic as a much more serious problem than generating a false alarm. Consequently, an IPS false positive is a much more serious matter than an IDS false positive. If an IPS blocks legitimate traffic for more than a few times, it will be pulled out of the network.

This difference changes the priority that a vendor places on false positives and puts significant burdens

on the filter writing and test teams. The IPS filter teams ask themselves not just, “will this filter detect the attack?”, but also “is there any way legitimate traffic could match this attack?”³ The test team extensively tests the traffic against terabytes of customer traffic, looking for matches of legitimate traffic.

The difference between IDS and IPS false positives also means that some types of filters that are appropriate for IDS are not appropriate for IPS. IDS filters create leads on suspicious activity intended for a human to follow. IPS filters are used for automatic action such as blocking traffic or quarantining an end point. The difference is suspicious activity vs. actionable intelligence. Any filter that detects suspicious activity, but might trigger a false positive, is appropriate for an IDS but inappropriate for an IPS. This means that most protocol or behavioral anomaly filters that are good for IDS are poorly suited for IPS because the intelligence is not actionable.

For example, an early filter written at HP TippingPoint detected a packet with the source IP address set to the loopback address. Such a packet should never be seen on a network, and is a protocol anomaly. When set to block, this filter shut down a customer’s network-based video surveillance system. The issue was that the cameras sent video data in UDP packets, with the source IP set to 127.0.0.camera-id.

³ When HP TippingPoint recruits a filter writer from a competitor, several weeks are spent indoctrinating it in the “no false positive” philosophy.

This real-world example illustrates a fundamental problem with using anomaly detection for automatic enforcement: legitimate traffic in real networks is riddled with anomalies. Protocol anomalies come from custom applications that use off-the-shelf protocol libraries, but use them in unexpected ways. Since the custom application is closed, and it works, the anomalous use of the protocol goes unnoticed until an IDS or IPS is installed. Behavioral anomalies come from exceptional, but often critical, business processes. For example, while testing a behavioral anomaly detector at a large e-commerce site, a huge amount of traffic was detected between two machines that had never communicated before. It turned out that the traffic was caused by a server crash that triggered an automated restore system to rebuild the server. The traffic of rebuilding the server was indeed a behavioral anomaly, but blocking it would have been disastrous.

These lessons, and others like them, demonstrate that anomaly-based detection mechanisms (both protocol and statistical) are useful for IDS, but inappropriate for IPS.

Some types of IDS false positives are not IPS false positives. If an IDS alerts on a real attack but the target is not vulnerable, it's a false positive. But if an IPS blocks that same traffic, it's not a false positive. In fact, blocking the traffic is beneficial since it reduces network load. Similarly, if an attacker crafts a stream to generate an alert on the IDS, that's a false positive. But when an IPS blocks the crafted stream, it is not a false positive.

When HP TippingPoint recruits a filter writer from a competitor, several weeks are spent indoctrinating them in the "no false positive" philosophy.

False positives create a potential vulnerability for the IDS. If a hacker can induce a false positive in an IDS, they can launch a "snow-blind" attack. In a snow-blind attack, hackers craft traffic that creates confusing alerts on the IDS console. Ideally, they hide all—the source, target and type of attack. While lighting up the IDS console, the attacker launches a real attack. The challenge for the security analyst is to somehow see the real attack amidst the chaff caused by the attacker. Alternately, if a hacker attempts to snow-blind an IPS with a crafted stream, the IPS will block the crafted packets, as well as the real attack.

The thrust of this section can be summarized in three rules:

- An IPS must have no false positives; an IDS "reduces" false positives. This dramatically changes the writing and testing of the filters.
- An IPS false positive blocks legitimate traffic; and IDS false positive alerts on an intrusion that did not—or could not—succeed.
- Anomaly filters cannot be used for blocking, only for alerting.

These distinctions change the requirements and engineering of the product, and make the IDS vulnerable to snow-blind attack.

Reduce false negatives: Network performance vs. decoding fidelity

A useful metric when evaluating IPS protection capabilities is to examine their coverage of important/critical Patch Tuesday vulnerabilities. Patch Tuesday is usually the second Tuesday of each month, on which Microsoft® releases security patches.

When push comes to shove and engineering trade-offs must be made, an IDS and an IPS will always be designed differently.

A false negative is simply a missed attack. Clearly a false negative is undesirable, and every vendor strives to provide as complete coverage as possible. However, there is no silver bullet: no product detects all attacks. Hence, the goal becomes providing coverage for high priority attacks. When prioritizing attack coverage, every vendor assesses three things:

- Is the attack important to customers?
- Can the engine do it without adverse impact?
- Is the filter writing team capable of researching and writing the filter?

A useful metric when evaluating IPS protection capabilities is to examine their coverage of important/critical Microsoft Tuesday vulnerabilities. Since these vulnerabilities are important to every vendor's customers, the vendor will prioritize development of these filters. The only reason not to provide coverage is if the engine is inadequate, or the team is incapable.

Besides lack of coverage, there are several other reasons for a false negative. The attack may

incorporate obfuscation techniques in order to evade the IPS or IDS. In the case of IDS, the IDS may be overwhelmed with traffic beyond its processing capacity and drop the packets needed to detect the attack. With an in-line device, overwhelming the device has a different effect: it causes traffic to be dropped. The attack does not succeed, since the attack packets are dropped, but it is also not detected.

In the case of evasions, both IDS and IPS should handle evasion tactics, but the way they handle them can be different. Ideally, both devices should unravel the evasion in order to correctly report the attack.

Consider an evasion that fragments MS-RPC traffic. The MS-RPC protocol natively supports fragmentation at the application layer. However, no legitimate application uses this feature on the vulnerable services; in fact, up until recent versions of Windows®, the MS-RPC fragmentation code was buggy. This bug went undiscovered for years, apparently because no one had ever attempted to use MS-RPC fragmentation in a legitimate application.

In this instance, an IPS could simply block streams that contain MS-RPC fragmented payloads instead of decoding the attack. Although it would be ideal to decode the attack, doing so would take processing cycles that would impact network performance (there is no free lunch). For an IPS, network performance is more important than decoding fidelity, provided there is no false positive. For an IDS, decoding fidelity is more important than network performance.

Note that this option, namely alerting on MS-RPC fragments, is not available to an IDS for other reasons. Alerting on fragmented MS-RPC traffic could be used to launch a snow-blind attack. Moreover, if an IDS alerts on fragmented MS-RPC traffic without decoding it, there is insufficient information to tell whether the attack was successful, whether the target may not have been vulnerable to the underlying attack, or if there was an attack in the payload. As a result, the IDS may generate a false alarm if it uses this method. A useful metric when evaluating IPS protection capabilities is to examine their coverage of important/critical Microsoft Tuesday vulnerabilities. When push comes to shove and engineering trade-offs must be made, an IDS and an IPS will always be designed differently.

IDS: Theory vs. reality

IDS vendors often confuse the issue of fundamentally different design requirements with theoretical discussions. Examples of this confusion are found in statements like:

- IDSs are based on protocol decoders vs. IPSs, which are based on pattern matching
- Whether or not a product can detect a certain anomaly (e.g., HTTP or encrypted traffic on a non-standard port)
- Whether or not a product is resistant to a particular evasion

They will utilize crafted traffic that causes false alerts and otherwise focus on minutia in order to divert the discussion from the basic mission of each product. These arguments often degenerate into ideological discussions.

The “protocol decoders are better” argument is a good example of these tactics.

Protocol decoders are simply state machines that decode structured network communication. Knowledge of the protocol’s structure is encoded into a state machine. Alerts are generated by comparing the values output by the decoder to “threshold” values (for anomaly detection), or by comparing them to the combination of values that trigger known vulnerabilities.

IDS vendors often confuse the issue of fundamentally different design requirements with theoretical discussions.

The advantage of protocol decoders is that they can detect protocol anomalies, something that is hard to do with pattern matchers; however, they can cause problems with product stability and deterministic performance, and are a source of false positives. Every protocol decoder is a little bit of C or assembly code, and code has bugs. If a vendor must create new protocol decoders or change existing protocol decoders to detect an attack, they are effectively releasing new versions of software every time a new filter pack is released—often on a weekly basis. This situation can occur because the attack occurs in a new protocol (the JPEG file vulnerabilities threw all the protocol decoding vendors for a loop) or because a

new type of vulnerability is discovered in an existing protocol. For instance, before the Nimda worm, Unicode decoding was not present in any product. Adding the Unicode decoding required a rewrite of the protocol decoder or the use of a pattern matcher.

Such frequent changes in software can cause product stability and performance issues. No matter how diligent the test team, code released on a weekly basis will have bugs. These bugs can show up as stability issues, performance issues, or even vulnerabilities, as evidenced by the Witty Worm⁴ or the Snort DCE/RPC preprocessing overflow.⁵

Moreover, protocol anomaly detection is a source of false positives. The values used for thresholds require major, site-specific tuning upon installation and ongoing tuning every time a protocol decoder is changed or released.

Protocol decoders are sometimes required in order to detect some attack types. But the vast majority of attacks can be unambiguously detected with no false positives using pattern matching technologies (IPS), especially if the pattern matcher is more powerful than vanilla regular expressions.

In practice, all products use a combination of protocol decoders and pattern matchers. So why focus on protocol decoders?

Conclusion

This paper describes many critical differences in the nature and priority of IPS and IDS requirements. For an in-line device, stability and performance are paramount. Even if an IPS had perfect detection capabilities and no false positives, if the latency was too high, throughput too low, or the product was unstable, then it simply doesn't matter. The device will never be deployed in-line.

Therefore, the prioritization of requirements for an IPS is, and must be:

- Stability
- Deterministic network performance
- No false positives
- Minimize false negatives

Table 1: Prioritized requirements of an IPS vs. IDS

Objective	IPS		IDS	
	Inline, automatic block	Priority	Out-of-band, human alert	Priority
Stability	<ul style="list-style-type: none"> • Crash is catastrophic-network goes down 	1	<ul style="list-style-type: none"> • Crash is annoying to security analysts who lose visibility-but no impact on network or apps 	4
Performance	<ul style="list-style-type: none"> • Processing designed for peak network load (Gbps) • Small memory buffers (μsecs of latency) 	2	<ul style="list-style-type: none"> • Processing designed for average network loads • Large memory buffers to absorb traffic bursts, creating seconds to minutes of latency 	3
Accuracy—False positives	<ul style="list-style-type: none"> • False blocks @ Gbps rates and thousands of filters kill applications 	3	<ul style="list-style-type: none"> • Burdens security analysts with chasing false alarms 	2
Accuracy—False negatives	<ul style="list-style-type: none"> • Preventing automatic blocking of good traffic trumps failure to detect anomalies 	4	<ul style="list-style-type: none"> • Missed anomalies may be missed attacks (information is power) 	1

Fundamental design of an IDS prevents it from ever being an effective inline, automatic blocking device at Gbps rates.

⁴ US-CERT: United States Computer Emergency Readiness Team—Witty Worm, March 22, 2004; <http://www.us-cert.gov/current/archive/2004/03/22/archive.html>.

⁵ US-CERT: United States Computer Emergency Readiness Team—Sourcefire Snort DCE/RPC Preprocessor Buffer Overflow, February, 19, 2007; <http://www.us-cert.gov/cas/techalerts/TA07-050A.html>.



For an IDS, the prioritization of requirements is:

- Reduce false negatives
- No false positives
- Deterministic network performance
- Stability

The IDS priority list in table 1 is the exact opposite order of the IPS list. These differences in priority have ripple effects that shape the definition of product requirements; their priority when making tradeoffs in the design of the products; the technology used to engineer products; and the organizations that build, test, and support them. When push comes to shove and engineering trade-offs must be made, an IDS and an IPS will always be designed differently.

HP TippingPoint avoids ideological discussions and focuses on the pragmatic issue of building a product to the prioritized requirements of an inline IPS, namely, stability, performance, no false positives, and reducing false negatives. It should be understood that HP TippingPoint uses protocol decoders if absolutely necessary, but leans towards technologies that have as much field experience as possible because they improve product stability and performance. HP TippingPoint believes this is consistent with the design objectives of a true IPS.

To know more how HP TippingPoint focuses on building a product to the requirements of an inline IPS, visit www.hp.com/networking/security.



Get connected

www.hp.com/go/getconnected

Get the insider view on tech trends, alerts, and HP solutions for better business outcomes

© Copyright 2011 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

4AA1-2338ENW, Created June 2011

