
Whitepaper

Next Generation Firewalls

Restoring Effectiveness Through Application Visibility and Control

Mark Bouchard, CISSP
Missing Link Security Services, LLC
www.missinglinksecurity.net



Table of Contents

Introduction	3
Factors	3
The Threat Landscape	3
The Application Landscape	4
Flaws	5
Poor Vision	5
Inadequate Coping Mechanisms	6
Insufficient Performance.....	6
Fixes.....	7
Application Inspection at the Core.....	7
Robust Application Identification and Inspection	7
Fully Integrated Threat Protection	8
Streamlined Policy Management	8
Hardware That Enables, Not Disables.....	8
Essential Capabilities for a Complete Solution	8
Summary	9

About the author: Mark Bouchard, CISSP, is the founder of Missing Link Security Services, LLC, a consulting firm specializing in information security and risk management strategies. A former META Group analyst, Mark has assessed and projected the business and technology trends pertaining to a wide range of information security topics for over 10 years. He is passionate about helping enterprises address their information security challenges. During his career he has assisted hundreds of organizations worldwide with everything from strategic initiatives (e.g., creating 5-year security plans and over-arching security architectures) to tactical decisions involving the justification, selection, acquisition, implementation and operation of their security and privacy solutions.

Introduction

Network security gateways are under siege. New threats are being launched faster than ever and are increasingly targeting application-layer vulnerabilities. At the same time user-centric and enterprise applications alike are taking advantage of commonly allowed communication ports and services to ensure their passage across security boundaries and to facilitate operation in the broadest set of networking scenarios. The result has been a steady erosion of the effectiveness of network firewalls and, consequently, the illumination of fundamental flaws in the initial design and subsequent modifications to these foundational elements of most enterprise security strategies.

This paper will first explore a combination of ongoing and emerging factors that are exposing a variety of deficiencies with current firewall designs. These issues and flaws will then be used to establish the criteria that define the ideal solution: a next-generation firewall that incorporates application inspection at its core.

Factors

There are a handful of trends impacting the effectiveness of conventional firewall products and ultimately driving the need for a next-generation solution. In general these trends break down into two categories: those related to the evolution of threats, and those related to the evolution of applications.

The Threat Landscape

Pertinent changes to the threat landscape stem primarily from a significant shift in hacker motivation. Simply put, hackers are no longer hacking to build their reputation. They now do it to make money. Consequently, it is no longer in their interest to devise threats that are fast and noisy, or that have benign payloads. Instead the name of the game is information theft. And this has led to two distinct avenues of pursuit. Let's call them the high road and the low road.

With the high road, the goal is not only to get away with the crime initially but to have it remain undetected for as long as possible, thereby preserving the value of the pilfered data and enabling repeated use of the same attack mechanisms. The result of these objectives has been a step-function increase in the sophistication of threats that are being generated. Rootkits, for example, are becoming more prevalent. These kernel-level exploits effectively mask the presence of other types of malware, enabling them to *persistently* pursue the nefarious tasks they were designed to accomplish (e.g., intercept keystrokes).

Targeted attacks are another major concern. In this case, hackers focus their attention on one organization at a time, often building customized attack mechanisms to take advantage of the specific equipment, systems, applications, configurations, and even personnel employed at a given location.

The increasing prevalence of application-layer attacks is yet another example of the growing sophistication of threats. For some time now hackers have realized that the majority of commonly deployed countermeasures are focused on providing network-layer protection. It is not a surprise, therefore, that greater than 80% of all new malware and intrusion attempts are exploiting weaknesses in applications, as opposed to weaknesses in networking components and services.

In contrast, purveyors of the low road eschew sophistication in favor of speed – speed of initial threat generation, speed of modification, and speed of propagation. The goal is to develop, launch, and quickly spread new threats immediately on the heels of the disclosure of a new vulnerability. The resulting zero-day and near-zero day exploits then have an increased likelihood of success because reactive countermeasures, such as patching and those tools that rely on threat signatures (e.g., antivirus software, intrusion detection systems), are unable to keep up – at least during the early phases of a new attack.

This speed-based approach is facilitated in large part by the widespread availability of threat development websites, toolkits, and frameworks. Unfortunately, another by-product of these resources is the ability to easily and rapidly convert “known” threats into ones that are “unknown”, at least from the perspective of signature-based countermeasures. This transformation can be accomplished either by making a minor

tweak to the code of a threat, or by adding entirely new propagation and exploit mechanisms, thereby creating what is commonly referred to as a blended threat.

In any event, the increasing speed and sophistication of threats emphasize the need for proactive, positive-model countermeasures with extensive visibility and control at the higher layers of the network computing stack.

Positive Versus Negative Countermeasures

Negative-model countermeasures operate on the basis of enumerating all communications and content that is known to be bad by virtue of its potential to cause damage.

Antivirus software and intrusion detection systems are classic examples of this type of tool. The challenge is that new threats cannot be stopped until they are identified and the tools are updated with the specific means to detect them (e.g., a signature).

In contrast, positive-model countermeasures operate on the basis of allowing all communications that are known either to be appropriate or necessary in a given situation, and then excluding everything else. The advantage is that such communications can be defined in advance, thereby enabling associated tools, such as firewalls, to automatically block a wide range of both known and unknown threats.

	Positive Enforcement	Negative Enforcement
Security Technology	Conventional Stateful FW	IPS/DPI Antivirus Anti-Spyware

The Application Landscape

The need for countermeasures with greater application awareness is also being driven by ongoing changes to the application landscape. User-centric describes applications designed primarily for personal communications, such as instant messaging, peer-to-peer file sharing, web mail, and IP voice/video collaboration. From an IT/security perspective, there are several challenges with this class of applications, including the following:

- A high degree of popularity assures their presence in the workplace, even when policies state otherwise. This is often facilitated by the ability to dynamically adjust their means of communication. Specifically, many of these applications can evade traditional security mechanisms by randomly shifting their communications ports and protocols, or by tunneling within other commonly used services (e.g., http, https).
- They are increasingly drawing the attention of hackers, both as a means for conveying malware and as a target in their own right. Indeed, the SANS Top-20 Internet Security Attack Targets for 2006 (November) has P2P file sharing applications and instant messaging listed as its tenth and eleventh entries, respectively.
- In many instances they can serve legitimate business purposes, not just personal ones. But if the available network and security tools are incapable of making sufficiently granular distinctions, then it comes down to an all or nothing proposition – either both objectives are supported, or neither of them are.

At the same time that organizations are wrangling with the above challenges, enterprise applications are presenting another, closely related problem. For a variety of business reasons – including easier development and deployment, potential cost savings, and improved accessibility – many client-server applications are being transitioned to take advantage of web technologies. Alternatively, these legacy applications are being replaced with hosted, web-based services, such as Salesforce.com, or the up-and-coming suite of productivity applications from Google. The issue is that these essential business applications are becoming indistinguishable from the plethora of less important applications that also utilize HTTP for the purpose of network communications. They cannot be controlled with distinct policies. Nor can they be treated to different levels of service.

The bottom line is that the prevailing conditions which characterize the application landscape are further reinforcing the need for security solutions with deeper awareness of and control over individual

applications. Without such capabilities a significant amount of unwanted, potentially threat-laden traffic will inevitably plague enterprise networks.

Flaws

Whether standalone or as a foundational component of increasingly popular unified threat management devices, firewalls are by far the most common network security solution deployed by IT organizations. Because firewalls are fundamentally positive-model controls, it is reasonable to expect that they would be able to address the issues discussed above. However, in practice most firewalls fall disappointingly short, primarily due to a combination of three factors: poor vision, inadequate compensating capabilities, and insufficient performance characteristics.

Poor Vision

As it turns out, most firewalls are far-sighted. They can see the general shape of things, but not the finer details of what is actually happening. In other words, most firewalls can infer common application-layer services from established port numbers (e.g., TCP port 80 is HTTP), but are unable to discern and therefore provide control over the individual applications that are using those services.

Indeed, two shortcomings with many firewalls that employ stateful packet filtering are: (1) that they assume a given application service is in use based on the tcp/udp port number that appears in a packet's header, and (2) that they only look at the first packet in a session to determine the type of traffic being processed. In general, these tactics are being used to enhance performance. However, they are flawed. The relationship between port numbers and applications is just a convention; adherence is not necessary for successful end-to-end communications. In addition, the first packet in a stream usually contains limited information. Only by examining subsequent packets is it possible to reliably establish the actual application and specific functions or commands that are being used. In practice what this means is that firewalls characterized by these shortcomings cannot:

- Properly account for applications that use non-standard ports, such as when a web server is running on a port other than those commonly associated with HTTP (i.e., 80 and 443), or when Yahoo! Messenger is running over TCP port 80 instead of TCP port 5050;
- Properly account for application tunneling, such as when P2P is file sharing or an IM client like Meebo is running within HTTP; or
- Provide granular control over individual functions, such as being able to block file transfers from within an otherwise allowed instant messaging application.

To be clear, the problem is not with stateful packet filtering per se; rather it is with specific implementations of the technology. Being stateful simply means that outbound sessions are tracked so that communication ports for return traffic can be dynamically opened as needed, versus being left open permanently. The deficiencies described above stem from how the sessions/applications are classified in the first place, which is a separate yet closely linked capability.

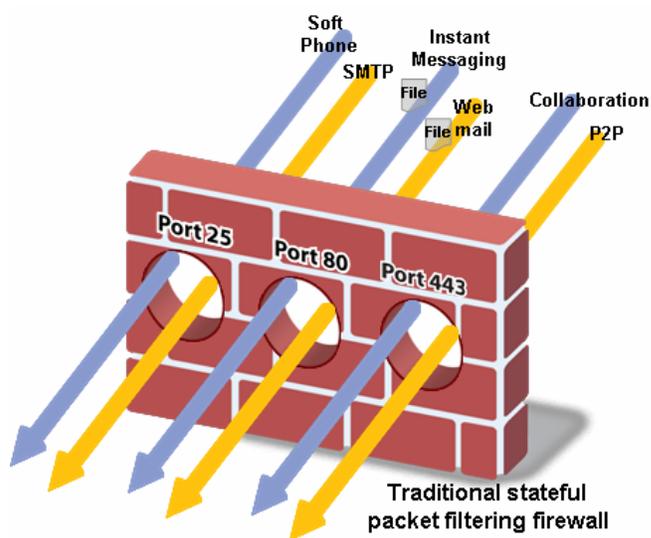


Figure 1: Stateful packet filtering is unable to identify certain applications because of their evasive techniques for communicating.

Inadequate Coping Mechanisms

To their credit, firewall vendors recognized the need to improve their products when the issue of threats migrating up the computing stack first began to emerge several years ago. Thus was born the concept and feature set now widely referred to as deep packet inspection (DPI). Unfortunately, despite the way it sounds, this approach does not involve inherently correcting a firewall's vision. Instead, DPI is based on compensating for poor vision by adding to the base product one or more negative-model security engines that already have a significant degree of application-layer coverage (e.g., antivirus, intrusion prevention).

In general, bringing additional countermeasures into the mix is a sound strategy. However, in this case the situation is somewhat analogous to building a house on a foundation of sand. At first it looks good. It even provides adequate shelter. But before long the critical error is revealed as the structure increasingly fails to withstand the elements to which it is exposed. Similarly, augmenting a firewall with DPI will always yield a boost in security effectiveness, but the degree of improvement will be limited in most cases because (a) the additional capability is effectively being "bolted on", and (b) the foundation it is being bolted to is weak to begin with.

Indeed, issues and limitations common to network security gateways that are essentially DPI bolted on to firewalls with poor application awareness include the following:

- Not everything that should be inspected necessarily gets inspected. Network traffic is only presented to the DPI engine if it matches a firewall rule *and* if that rule has the attribute associated with further threat inspection set to "on". In this regard, DPI can provide a false sense of security, since risky applications can pass unchecked by tunneling within services that are generally considered safe.
- Any other security engines that are part of the overall system (e.g., antivirus, anti-spyware) are subject to a similar situation. Again, the base classification engine remains the firewall. So despite the fact that the DPI engine has application awareness this better information is not reliably available for use as the basis for triggering additional inspections.
- Policy management can become convoluted. Ideally security tools have one policy table to control flows and another one to specify what is done when a threat is detected. However, in this case providing more granular flow control depends on "nesting" access control policies within the threat prevention portion of the product – which itself is only engaged as part of a higher level access control policy.
- System resources are potentially used inefficiently. This issue applies primarily when the firewall solution incorporates multiple, separate threat detection engines that rely on fundamentally similar inspection techniques (e.g., antivirus, anti-spyware, and DPI). In such cases, the amount of redundant processing that occurs can be considerable.

Of course, redundant processing only exacerbates the third factor that impacts the effectiveness of current firewall products: insufficient capabilities to ensure adequate performance.

Insufficient Performance

Regardless of how it is technically accomplished, providing granular application awareness is a CPU and memory intensive process. So unless a firewall system has been designed from the outset with application-layer inspection as an objective, then it will usually exhibit performance issues. As a result, choices – also known as compromises – will have to be made. To ensure adequate performance levels can be sustained, organizations will need to selectively implement the application-layer inspection features, as well as any other advanced filtering capabilities for that matter. Indeed, the impact on performance is one of the primary reasons why DPI, when it is bolted on, is simply not "on by default" for any of the firewall rules.

To be clear though, just having application-layer inspection as a stated design goal is by no means sufficient. For instance, some vendors will try to make due by throwing bigger and faster off-the shelf hardware at the problem. This will always work to some extent, but is typically far from optimal and can be quite expensive. In contrast, a customized hardware architecture that applies the right balance of

computing resources for the tasks at hand provides greater assurance that rated performance levels will be achieved.

Fixes

The previous paragraph begins to shed some light on the features and capabilities that characterize a solution that overcomes the deficiencies with many of the firewall products currently available in the market – a process which this section of the paper is intended to complete. Overall, to be effective a next-generation firewall must include and account for application insight at the core of its design, while also addressing the full range of capabilities associated with a robust, network-based security platform.

Application Inspection at the Core

Although it yields a promising combination of both positive and negative-model countermeasures, bolting DPI on to a half-blind firewall is clearly less than ideal. A far more effective approach is to start with a firewall foundation that has better vision from the outset. Powerful positive-model techniques can then be used to a much fuller extent, prior to invoking negative-model threat inspection capabilities to ensure that allowed sessions are indeed free of threats. As always, however, the devil is in the details.

Robust Application Identification and Inspection

The goal is to control the flow of sessions more granularly, on the basis of the specific applications that are being used, instead of just the underlying set of often indistinguishable network communication services. Reliably meeting this objective depends on taking a multi-factor approach to application inspection and identification. Establishing port and protocol is a first but insufficient step, as noted earlier. Considerably more important is the presence of an extensive library of application signatures *and* the means to apply them.

A signature in this case is essentially a set of telltale signs that characterize an application and, to the extent possible, definitively distinguish that application from all others. In terms of applying signatures, this entails having the ability, including the necessary system resources, to conduct advanced inspections, such as: looking beyond the header and into the payload of individual packets; looking beyond the first packet in a given session; and, perhaps even re-building portions of a session to enable higher-level analysis. Notably, for “allowed” sessions, advanced inspections and application decoding will need to be conducted on a continuous basis: (a) to enable enforcement of application-specific policies (e.g., preventing file transfers from an otherwise allowed instant messaging application), and (b) to facilitate accurate and potentially more efficient use of available threat signatures, which are discussed in the next section.

One final consideration in this area is the ability to handle encrypted traffic, a particularly relevant concern given that SSL-protected applications are becoming increasingly prevalent. The issue is that conventional firewalls cannot inspect such traffic to any meaningful extent. They are limited to enforcing policy based on a few tidbits of information available in the packet headers (e.g., source and destination IP address). To alleviate this blindness, a next-generation firewall should ideally be equipped with the ability to optionally decrypt and re-encrypt SSL-protected traffic, thereby enabling it to conduct granular access control and a more thorough inspection for embedded threats.

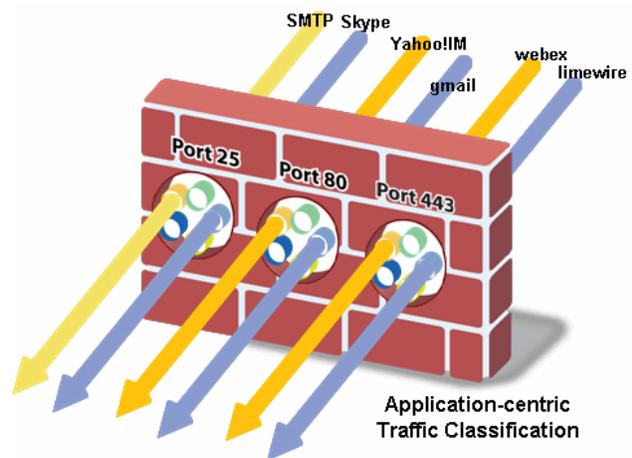


Figure 2: Application-centric traffic classification identifies specific applications flowing across the network, irrespective of the port and protocol in use.

Fully Integrated Threat Protection

Just because a session is allowed by policy does not mean it is free from threats. This is true regardless of the granularity with which the policy is enforced, and is why it makes excellent sense to also bring detailed threat inspection capabilities into the mix. However, simply adding a series of separate threat protection engines to a firewall is a recipe for trouble. Although this approach should improve security, it will inevitably cripple the system from the perspective of performance, forcing organizations to strike a balance between the two.

In contrast, having fully integrated threat protection significantly reduces the performance burden, thereby eliminating the need to compromise on the security front. With this approach there is only one threat protection engine, instead of having one each for antivirus, anti-spyware, intrusion detection/prevention, and so forth. As a result, fewer hand-offs are required between internal processes and, more significantly, the packets are “cracked” (i.e., processed) less often. Of course, a prerequisite to having a single engine is having a standard signature format. Ultimately, this too should prove beneficial to organizations since it reduces the knowledge set required for routine management functions as well as the development of customized signatures.

Streamlined Policy Management

Having application inspection at the core of the firewall results in a more intuitive policy model than when it is bolted on. Access control rules can be implemented much more efficiently and with less likelihood of introducing errors since they can now be contained in one place in the management interface. For streamlining purposes, another valuable feature is the ability to implement, with a single configuration setting, a set of default responses for the subset of detection signatures that definitively correlate with the presence of a threat.

Hardware That Enables, Not Disables

Much of this topic has already been discussed. Overall, it should not be necessary to make compromises, in the form of selectively implementing available security features, to maintain adequate levels of performance. Rated throughput and reasonable latencies should be sustainable even when all application and threat inspection features are engaged simultaneously – which is the ideal configuration from a security perspective. In this regard, it is typically not a good sign if a firewall product is implemented with run-of-the-mill server hardware. That said, the intent is not to imply the need for specialized silicon (i.e., ASICs). Instead, the critical aspect to confirm is the presence of a customized hardware architecture. For example, features such as maintaining separate data and control planes and the use of merchant silicon (i.e., dedicated, yet off-the-shelf chips for accelerating specific processes) are just two telltale signs that the vendor has made a concerted effort to build a solution with sufficient resources to ultimately deliver top-notch performance.

Essential Capabilities for a Complete Solution

Of course, resolving the flaws and deficiencies with current firewall designs is really just the starting point for a next-generation solution. The impact of having application awareness at the core, integrated threat protection, and streamlined policy management will be significantly diminished unless these crucial enhancements are enabled as part of a truly enterprise-class platform. Beyond having high-performance hardware, other essential characteristics and capabilities of such a platform include the following:

- Networking flexibility helps ensure compatibility with virtually any organization’s computing environment. Enabling implementation without the need for re-design or re-configuration depends on supporting a wide range of networking features and options, such as 802.1Q and port-based VLANs, trunked ports, transparent mode, and numerous, high-capacity interfaces.
- Reliability helps ensure non-stop operations and entails features such as active-passive and/or active-active failover, state and configuration synchronization, and redundant components (e.g., dual power supplies).

-
- Scalability is primarily dependent on having solid management capabilities and high-performance hardware but can also be facilitated by support for virtual systems, where one physical firewall can be configured to act as many.
 - Finally, manageability entails the over-arching characteristic of being easy to use. It also involves capabilities such as local management, centralized management, role-based administration, automatic signature updates, real-time monitoring for both device status and security events, and robust logging and reporting.

Summary

The firewall is a cornerstone of most organization's information security strategy. However, the effectiveness of this security stalwart is steadily diminishing as threats continue to migrate up the computing stack and as applications of all types are engineered to take advantage of web technologies and other services that are typically allowed by enterprise policies. Furthermore, attempts to counteract this trend by bolting capabilities such as deep packet inspection on to conventional firewall products are not sufficient. Too much unwanted traffic, some of it potentially laden with threats, is still able to get through. What organizations need instead is a next-generation firewall system – one that incorporates application awareness at the core of its design, has fully integrated threat protection, and also includes a customized hardware architecture to avoid the need to choose between security and performance.