White Paper

Juniper's Deep Inspection Firewall

Making Application-Level Attack Protection Pervasive

Sarah Sorensen
Product Marketing Manager

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
408 745 2000 or 888 JUNIPER
www.juniper.net

Part Number: 200050-001

Contents

Introduction

As data-driven, application-layer attacks have proliferated in recent years, it has become increasingly clear that the existing solution set is not adequate to counter these threats in a cost-efficient manner. Implementing effective security to address the myriad of threats aimed at an organization's network is difficult without the right tools. The primary tool that most enterprises use to protect their networks is a Stateful inspection firewall. While effective at enforcing access policies and protecting against exploits at the network level, such as Denial of Service attacks, these solutions are not designed to protect against intrusions that target the application. It is these application-level attacks that are making headlines and costing enterprises millions of dollars in lost intellectual property, revenue and productivity.

In 2002, intrusion prevention systems (IPS) were introduced to provide application-level attack protection, inspecting and then preventing the attacks in the traffic allowed by the firewall. The early adoption of intrusion prevention technology has mainly been focused on securing sensitive resources in the corporate headquarters and large regional offices. Generally, customers deploy IPSes behind a Stateful inspection firewall and in front of critical servers, where protecting application-level data from both internal and external attacks is a primary concern.

NOTE For more information on Intrusion Prevention Technology, please see Juniper Network's white paper "Intrusion Detection and Prevention: Protecting Your Network from Attacks."

It is important to deploy this application-level protection throughout the network, since an organization's ability to protect any network resources can be compromised by a single "weak link." Small remote and branch offices and telecommuters with home networks also need application-level attack protection. Recognizing that these network segments probably do not have the resources or variety of protocols running through them that the regional offices and large central sites do, the need is to add the appropriate application-level protection for the resources found at these sites, i.e. a Web server, e-mail server, etc.

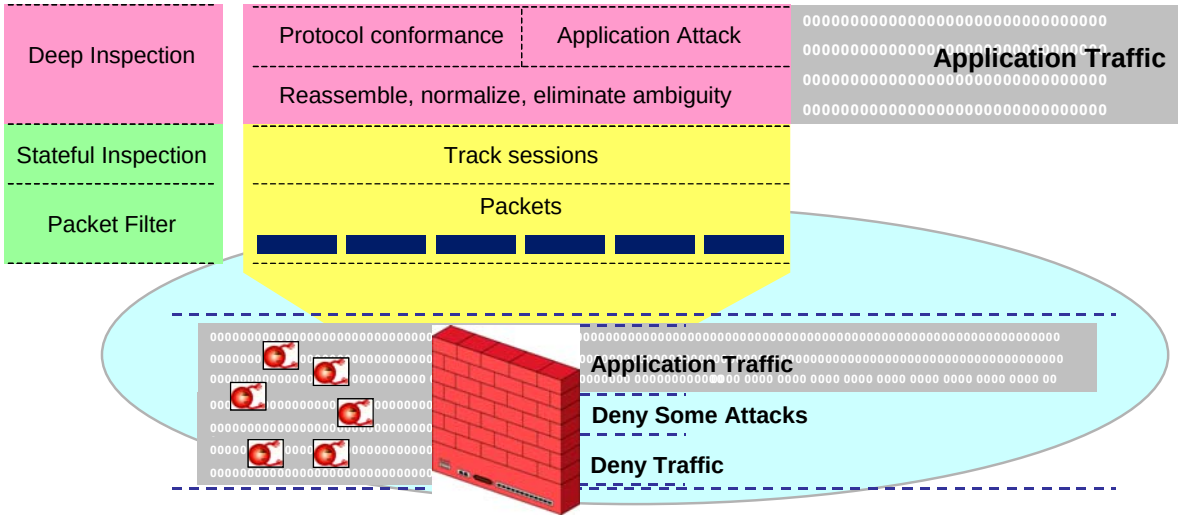
The logical place to add this protection is at the perimeter of these smaller network segments, where the firewall usually sits. Adding deeper protection to the firewall for the types of attacks that threaten these network segments would enable the organization to prevent these threats at the edge and strengthen their overall security. Such a strategy would meet the security goals at minimal cost to management overhead and complexity. This is what Juniper Network's Deep Inspection (DI) firewall is designed to provide, enabling enterprises to easily deploy application-level attack protection throughout their extended network to mitigate the risks posed by these traditionally "weaker" links.

Juniper's DI firewall represents the next generation in firewall technology. This paper describes the implementation of Juniper's DI firewall, explains how DI protects against application attacks, demonstrates how customers can control and manage the application-level protection, and discusses various deployment strategies to illustrate how to most effectively use DI firewall.

NOTE For more information on traditional technologies and the requirements for application-level protection, please read Juniper's white paper "The Need for Pervasive Application-Level Attack Protection: How Deep Inspection Technology Meets the Requirements for Network and Application Attack Protection."

Juniper Network's Deep Inspection Implementation

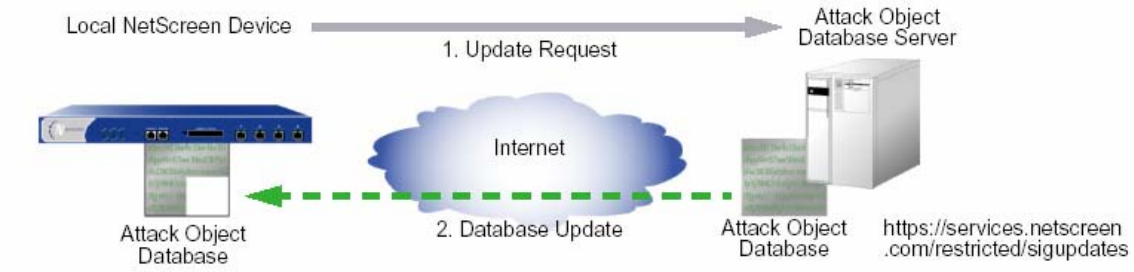
Juniper's Deep Inspection firewall provides both network and application-level protection to strengthen the overall security stance of the network. The DI firewall integrates the strengths of both Stateful inspection and intrusion prevention technologies to efficiently process network traffic and insulate network resources from many of the sophisticated attacks targeting Internet protocols. With all of the benefits of Stateful inspection, the DI can quickly perform network-level analysis to make access control decisions on the traffic and then, for the traffic that is accepted, look deeper in the traffic to make additional decisions based on the application-level information.



Juniper's Deep Inspection firewall is designed to provide the application layer protection for the most prevalent Internet-facing protocols, such as HTTP, SMTP, IMAP, POP, FTP and DNS, with the ability to easily add protocols in the future. For these protocols, the Deep Inspection Firewall interprets application data streams in the form that a recipient would act upon. To accurately interpret the intent of the traffic at the application level, the DI firewall performs de-fragmentation, reassembly, scrubbing and normalization. Once the DI firewall has reconstructed the network traffic, utilizing the aforementioned processes, it employs protocol conformance verification and service-field attack pattern matching to protect against attacks within that traffic.

The DI firewall utilizes an Attack Object Database to store protocol anomalies and attack patterns (sometimes referred to as signatures), grouping them by protocol and security level (severity), to perform both its Stateful Inspection and Deep Inspection duties. The firewall's analysis engine extracts the relevant attack objects that it needs from its local database at runtime to effectively analyze the traffic.

The initial release will protect against over 250 application attacks. Of course, the Attack Object Database may be updated, either automatically or manually, as depicted in the figure below.



The update process entails connecting to the Juniper Attack Object Database Server, which will be updated regularly with new attacks and updates to preexisting attacks. Emergency updates will also be provided to protect against key attacks as necessary. Customers must have a software subscription and subscribe to the Target Attack Update Service to receive these new attack protection enhancements.

How The Deep Inspection Firewall Protects Against Attacks

Once Juniper's Deep Inspection firewall reconstructs the packets on the network to the application message, it then performs application specific analysis to determine whether the intent of the traffic is malicious or benign. First, it analyzes the data based on the protocol specifications. If the data deviates from the specifications, it represents an anomaly. High-impact anomalies that represent malicious intent, such as trying to overflow a memory buffer to take control over a system, are identified as attacks. Granular control of how and where to look for anomalies allows non-conforming systems to be supported. The DI then uses its knowledge of the protocol to identify the application communication service fields and perform specific pattern matches against the relevant fields to identify attacks. Service Fields are portions of the traffic that relate to specific functions, such as email addresses, URLs, file names, etc., that the DI firewall applies attack pattern matches to in order to identify an attack. These service fields represent the application message and enable the DI firewall to understand the communication to look for attack patterns in the right fields.

Protocol Conformance Verification for Day-Zero Defense

Because Juniper's DI can apply protocol conformance to the traffic, it is able to protect against entire classes of exploits, such as buffer overflow attacks, without needing to know about a specific exploit. This means that the solution can potentially provide "day zero" protection against brand new attacks as they emerge. It is also a mechanism that can protect against some of the more sophisticated attacks that cannot be characterized by a simple pattern.

The DI firewall's analysis engine compares the message content with the RFCs that govern protocol behavior – that is, comparing the format of the transmitted protocol with the standards specified in the RFCs and RFC extensions for that particular protocol or in Internet Drafts and source code, when applicable.

Let's look at exactly how the DI firewall uses protocol conformance verification to protect against attacks, using a "directory traversal attack" as an example. With IIS Web servers, the default location for public web server files on the Microsoft file-system is C:\INETPUB\WWWROOT. A directory traversal attack, targeting the Microsoft Windows operating system running IIS, will attempt to fool the web server into allowing clients to access files and system resources outside of WWWROOT. The following HTTP GET request is an example of an attacker who has sent a deliberately crafted message designed to fool the host machine into granting access to critical resources to the client:

```
GET
/USsales/revenues/../../../../WINNT/SYSTEM32/CMD.EXE?/C+dir+C:\+/S
```

Here the attacker has used the GET command to gain access to the cmd.exe program, which is the access point to the Windows shell. Instead of a CGI program, the URL contained within this HTTP request directs the IIS server to invoke CMD.EXE. The DI firewall, upon receiving and parsing this message, will transform the original application layer message shown above into the form that the host machine will see:

```
\WINNT\SYSTEM32\CMD.EXE /C dir C:\ /S
```

This command issues a directory listing of the entire hard drive, and the IIS server will respond by passing this information back to the client in the HTTP response. Without Juniper's DI firewall to stop this attack before the Web server processes this command, the attacker will have succeeded in using a malformed URL to breach network security.

To get to the point where the DI firewall is able to analyze the HTTP GET message to detect and stop the security breach, the firewall must reconstruct the application layer traffic. The first step is to reassemble the TCP packets belonging to this particular communication session back into their original sequence. If the original HTTP GET command, after having been encapsulated as a TCP stream, is fragmented into 10-byte packets the message will be transmitted as shown below:

Ideal transmission of malicious HTTP GET message

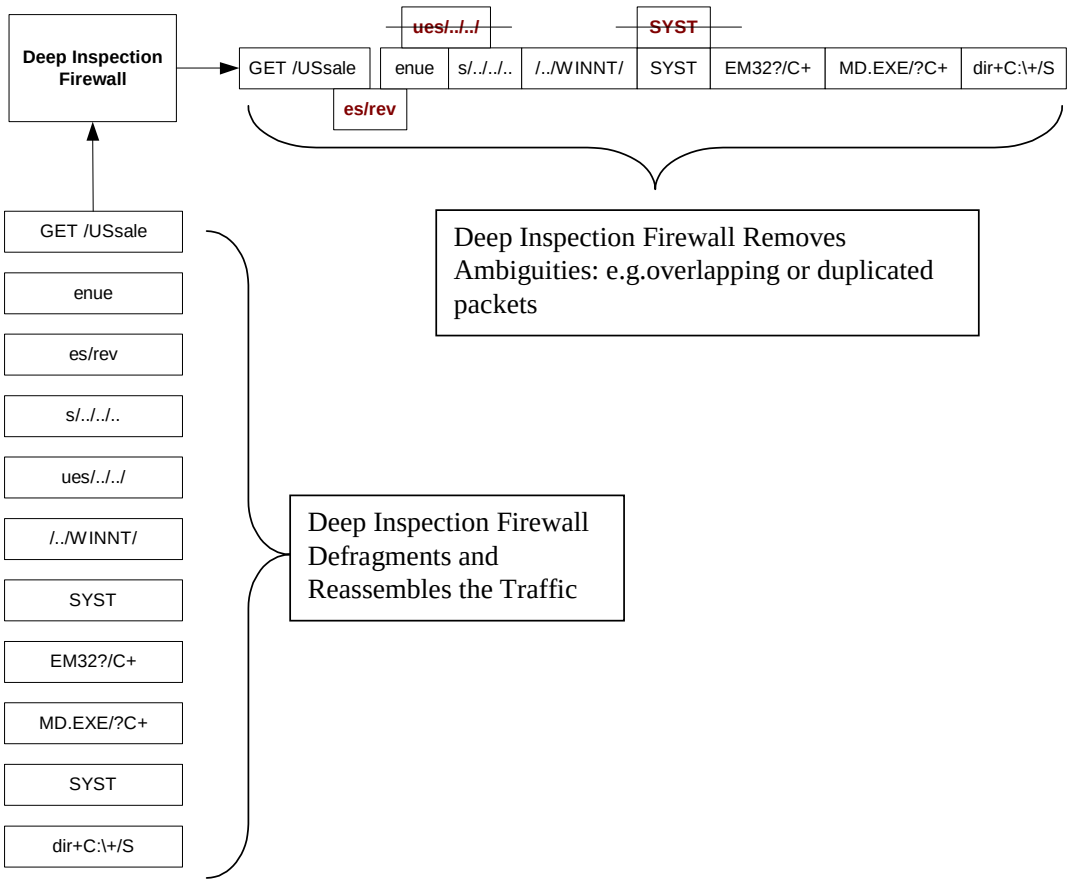
GET /USsales/revenues/../../../../WINNT/SYSTEM32/CMD.EXE?/C+dir+C:\+/S

Transmit Order	Packet Payload
1	GET /USsal
2	es/revenue
3	s/../../..
4	/../WINNT/
5	SYSTEM32/C
6	MD.EXE?/C+
7	dir+C:\+/S

Of course, this is an idealized scenario. Often the host machine, and by extension the DI firewall, will receive the packets out of sequence. The problem is compounded due not only to network infrastructure, which can force packets to be further decomposed and re-sequenced, but also attackers who deliberately alter packets in order to try to fool security solutions. For example, the intruder could force packets to be needlessly fragmented at the IP

layer (possibly into different sizes), transmit them out-of-sequence, send duplicate packets, overlap content across packet boundaries, or any combination of the above. It should be noted that all of these occurrences could very well occur even without an attacker attempting to breach the network, just due to the vagaries of the network's infrastructure. Thus, the DI firewall can handle a potentially scrambled message stream, resolve the ambiguities present within the data stream, and reconstruct the message in order to analyze it's content.

An example of such scrambled network traffic, using the same HTTP GET message is shown below:



The DI firewall, upon caching the received packets, will proceed to analyze the TCP sequence numbers and reorder the packets such that they appear in their original order. The above example illustrates how performing the TCP reassembly is necessary, but not sufficient, in order to analyze the message content, for the reassembled network packets cannot be used for analysis in their raw form. The DI firewall's primary goal is to treat the message content in exactly the form that the host machine will interpret the data, and this entails resolving the potential ambiguities that arise from actual network transmission.

The ambiguities will be resolved during the normalization phase. For example, in the above example notice the duplicate packet with the payload "SYST" – clearly the duplicate must be flagged and dropped from the message stream. In addition, there may be packets with overlapping data, like the packets that contain "es/rev" and "ues/../../". In the case of the "ues/../../" packet, it must be dropped because it contains overlapping data that is duplicated amongst other packets. However, the other packet containing overlapping data

contains message content that must be preserved in order to reconstruct the original HTTP message. Through careful analysis of the offset value contained in the IP headers, the DI firewall is able to transform the fragmented content into packets that are temporally adjacent and contiguous. This comparatively simple example illustrates the importance of resolving the ambiguities present within the individual packets – without this ability no firewall has any hope of catching this directory traversal attack. Finally, the normalization phase also handles different encoded representations – for example the application layer message may be transmitted in unicode format instead of ASCII text, and hence the firewall will transform unicode-encoded text into ASCII before moving onto the final analysis stage.

Stopping this attack required a different technique than that of the IIS DoS attack. An attack signature that contains the string "/../../../../" will not work, as there are URLs where this string is valid (referencing image files in HTTP is one such case) – thus one would be susceptible to numerous false positives should the pattern matching approach be employed here. Similar examples include referencing the path /winnt/././ which is the same as /winnt, and relying on a signature approach here would be inviting trouble as an attacker could obfuscate the path statement in countless other forms, or even replace the slashes with their Unicode equivalent in order to evade a pure signature scanning approach. However, because the DI firewall is able to interpret the application layer message content as the host machine will be, the system is able to use analytical techniques to flag patterns of use that invalidate a protocol's intended use.

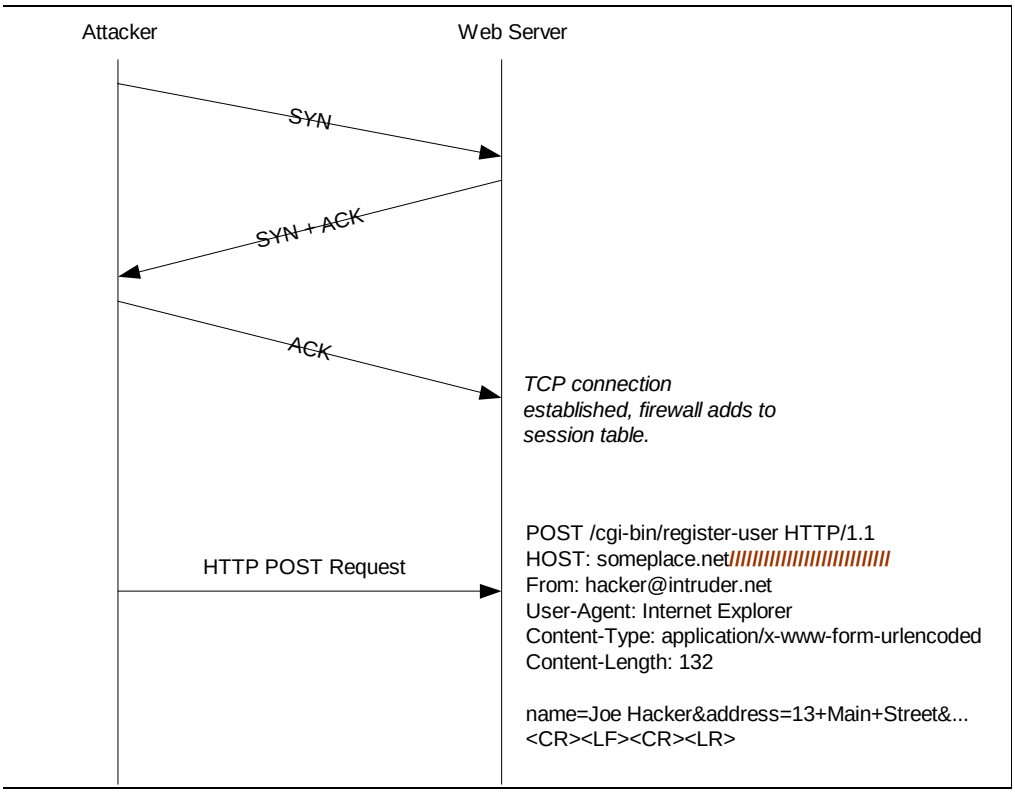
Service Field Attack Pattern Matching for Known Attacks

While protocol conformance verification is able to protect against some unknown and more sophisticated attacks, there are many attacks out there that are known and most efficiently protected against by matching known attack patterns to relevant areas of the traffic. Juniper's DI firewall performs application analysis to understand the intent of the message and then maps different portions of the traffic to their appropriate service fields. Service fields are pre-defined header values that come from the definition of the protocol in question and represent the intended purpose/use of that specific place in the traffic. For example, in SMTP, some of the service fields are: command-line (a command from the client to the server), data-line (a line of the e-mail message itself), From: (sender's e-mail address), etc. The DI firewall then applies the known patterns, which correspond to predefined "Attack objects" that reside in a database contained within the device, to the relevant portion of traffic where a match can do damage. Juniper's pattern matching conserves resources, focusing its energies on the relevant traffic where attacks are perpetrated, to efficiently protect against known attacks.

As an example of how service field attack pattern matches (sometimes referred to as Stateful Signatures) can protect network resources, consider the case of preventing against an attack that exploits a known vulnerability in the Microsoft Internet Information Services (IIS) Web server. Like many other Denial of Service (DoS) attacks, this attack uses deliberately malformed data, which due to an implementation bug, causes the system to slow down, with 100% CPU utilization. If the Microsoft IIS Web server [version 1155.0, 5.1] encounters a series of forward slashes inside of the "Host:" header of an HTTP request, it will crash. The client machine initiates communication with the web server by establishing a TCP connection to the web server using the three-way SYN/SYN+ACK/ACK handshake. A Stateful inspection firewall is generally configured to allow incoming connections on TCP port 80 and will let the connection through. The client machine has now established a TCP connection to the Web

server and the connection is added to the firewall's state table.

Now, the HTTP client sends an HTTP request message to the Web server. HTTP request headers are in RFC-822 format and contain the specific HTTP query (e.g., GET, POST, or HEAD), along with header lines containing the header data and finally an extra carriage return and line feed indicating the end of the request. The POST HTTP request specifies an URL indicating a CGI program to run, along with some other data that the web server will process by invoking a CGI program. Consider the following diagram, which depicts client/server communication where a malicious client has corrupted the "Host:" portion of the HTTP request, which if not detected and stopped will crash an IIS web server:



A Stateful Inspection firewall's job is done after it adds the TCP connection to its state table. It cannot search for the offending series of slashes because it is only designed to analyze information contained in the raw IP/TCP/UDP/ICMP/etc. datagrams – it does not search the payload of these datagrams, other than to look for auxiliary flows, such as those required by FTP or multi-media protocols. Thus, without more information the firewall has left the IIS web server open to attack. In contrast, Juniper's Deep Inspection firewall is capable of analyzing the HTTP protocol content in real-time and, as such, is able to identify the attack pattern and drop the malformed HTTP request. The security manager can direct the DI firewall to use the following regular expression to search through every host field in each incoming HTTP request:

```
Pattern: .*//////////////////////.*
```

The DI firewall will identify the attack by comparing the data contained within the application service fields with the Attack Object Database's library of signatures. For this particular HTTP example, some of the service fields the DI firewall will identify are the "HOST:," "From,:" "User-Agent," "Content-Type," and "Content-Length" portions of the HTTP request. An attack response will only be triggered if a series of slashes appears in the HOST service field. The DI firewall will not look for or trigger an attack response if the pattern is anywhere else in the HTTP request message, since the vulnerability only manifests itself if the pattern appears within the "Host:" field.

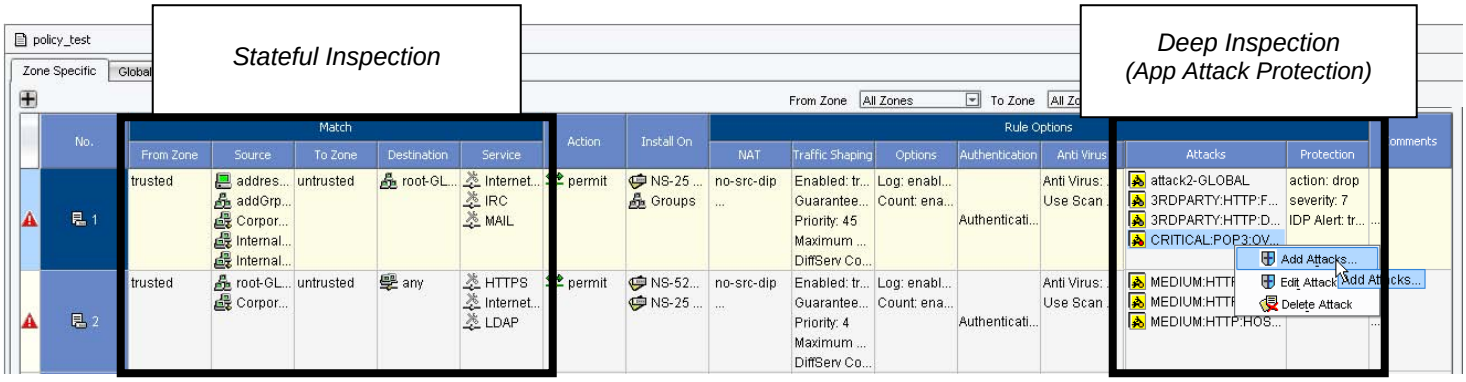
Juniper uses regular expressions to enable the DI firewall to protect against multiple possible representations with a single attack pattern match. Juniper's use of regular expressions to specify attack patterns provides a means by which administrators can easily add additional Attack Objects as new attack patterns become known. This flexibility and configurability are important, as this allows administrators to customize the firewall's capabilities to suit the needs of the organization. For example, if an organization's Web site does not use the POST command, then a signature that flags incoming POST commands can be used to stop any such suspicious activity. Of course, Juniper's in-depth knowledge of application layer content and ability to perform service field classification is a prerequisite to such functionality, since the string POST could easily appear in the body of an e-mail, in an attachment, or in countless other places. Organizations can pinpoint, using the given service fields, where in the traffic to look for their customized attack pattern matches for the protocols supported to efficiently protect their network resources.

Granular Control Over Juniper's Deep Inspection

It is important to be able to tailor any security solution so that it meets the unique needs of each organization. Juniper's Deep Inspection firewall was designed to give customers granular control, while simplifying the deployment, management and ongoing maintenance of the solution. Juniper offers customers the flexibility to interact with the Deep Inspection firewall via a command line interface, Web UI, or the Juniper Networks NetScreen-Security Manager. All three of these methods perform essentially the same task, that of setting up the analysis engine to perform whatever the organization needs the device to do. Combined with a flexible analysis engine, Juniper gives administrators the functionality necessary to deploy Deep Inspection effectively throughout the enterprise network.

It is of fundamental importance to ensure that you are applying application layer protection at the appropriate places, in the most efficient manner, per the network's topology. An office with an FTP server, Web site, SMTP gateway and some VPN users will require a different configuration than an office that only has a Web site. So it is important to have the flexibility to apply Deep Inspection to only the relevant traffic that poses a risk to that network segment. For example, in a remote office, where outgoing traffic is "trusted," an administrator may opt to protect just the incoming traffic. Or, if this remote office doesn't have an e-mail or Web server, then the administrator doesn't need to inspect the traffic for attacks in those protocols.

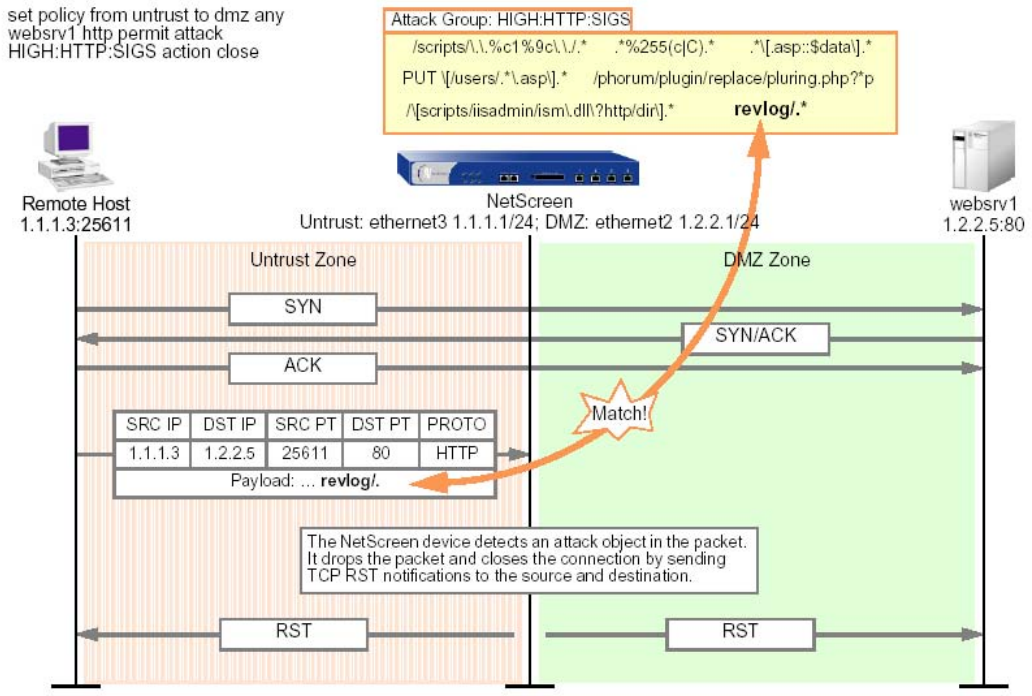
Juniper offers administrators complete control over exactly how the Deep Inspection capabilities are applied. Juniper does not force an administrator to make an all or nothing decision. With Juniper's DI firewall, administrators have the flexibility to choose what traffic to apply Deep Inspection to, which attacks to look for in that traffic- from a single attack to a group of attacks-, and where in the network to apply that rule.



The above screen shot of NetScreen-Security Manager shows this process in action. There are two policies shown here, illustrating how the firewall is configured on a per-rule basis. The DI firewall policies are rule-sets that allow an administrator to control, first, what traffic is allowed in and out of the network – stateful inspection - and then how Deep Inspection is applied to that traffic. Individual attack objects and groups can be toggled on or off and can also be edited to provide granular control over the application attack protection. Moreover, administrators can add new customized attack objects to their policies, applying attack pattern matches to relevant service fields for certain protocols.

Customization of the attack objects in the attack object database means the tradeoffs between Deep Inspection and just Stateful Inspection will be managed in the most effective manner possible. When Deep Inspection is added to a policy, the Juniper device will inspect the application message in the network traffic that the policy permits for any patterns matching those in the reference attack object.

The figure below shows this process in action:

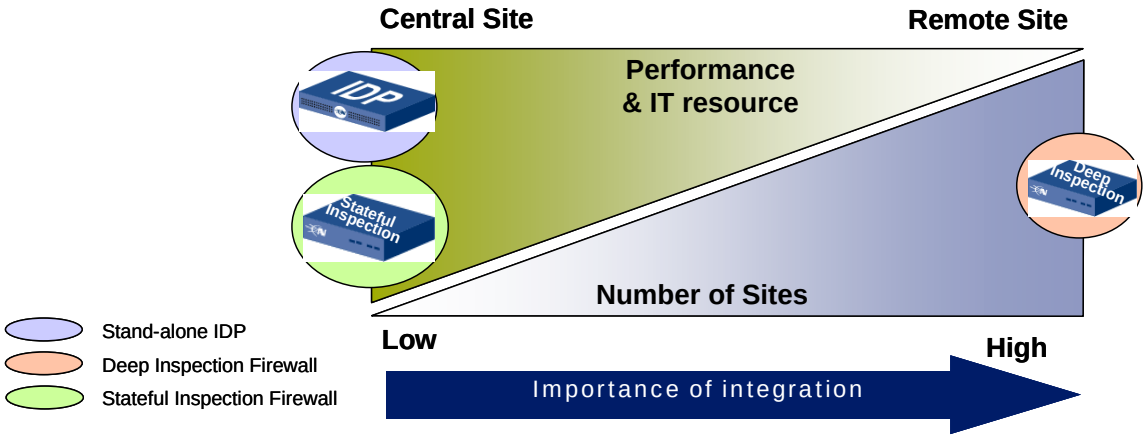


When a match occurs, an attack response will be taken depending on the instructions contained in the applicable policy. There are seven possible responses, ranging from just logging to actively closing the session between client and server. This particular HTTP permit attack, which is reminiscent of the HTTP post attack described earlier, shows how the DI firewall's analysis engine uses this information to construct a regular expression that is applied to the HTTP message data, in order to thwart an attack against the Web server. By giving the end user the ability to define exactly what attacks to look for and how to respond when those attacks are identified, DI firewall makes constructing robust network protection strategies feasible and intuitive.

Deployment Strategies

Securing a heterogeneous enterprise network from application-layer attacks requires both advanced protection devices, along with the flexibility to configure such devices to meet your specific needs. Flexibility and increased application-layer protection must be weighed against maintenance costs, but by moving Deep Inspection into the firewall Juniper has minimized this impact. IT administrators can utilize various Juniper products to implement a security strategy that matches their particular network topology, thus enhancing the security stance throughout the entire network. One technology that has not been covered but is an essential component of any application-layer protection strategy is antivirus. Antivirus is a complementary technology that looks for attack pattern matches in files, while Deep Inspection looks for attack patterns in the application message. Through its partnership with Trend Micro, Juniper offers an integrated AV solution for some of its products. This section will cover specific deployment strategies, and show how to pick and choose amongst the various devices to maximize security while minimizing maintenance costs.

NOTE For more information on the differences, please see Juniper's "Comparison of Firewall, Intrusion Prevention and Antivirus Technologies" white paper.



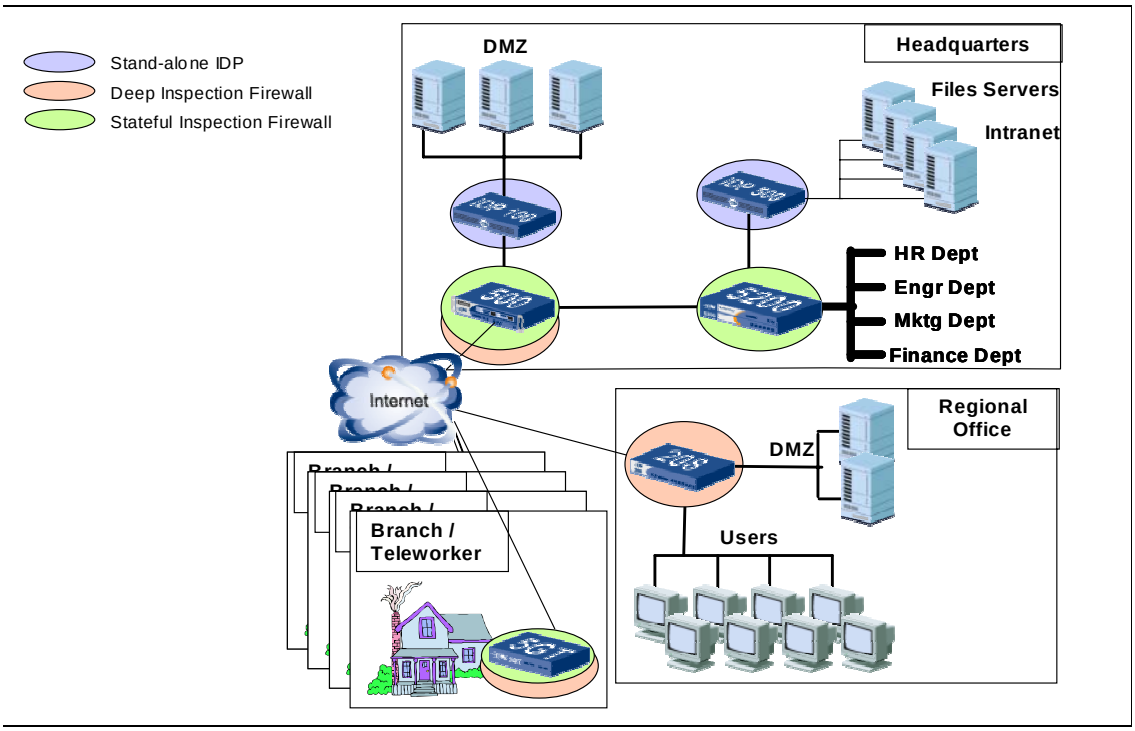
A typical enterprise network will be comprised of many nodes, varying in size from small branch offices or telecommuters to large centralized sites with many servers and internet gateways. The above graphic illustrates the interplay between network infrastructure, IT maintenance, and the need for integration in order to attain robust application-level protection.

At central sites and large regional offices, the need for scalability, performance, and total application-layer protection is tantamount. These sites generally have a lot of different resources, representing a lot of different applications and users. A typical configuration at a central site or headquarters will most likely include deployment of a Stateful Inspection firewall in tandem with an IDP device, as this particular strategy meets the need for scalability, sophisticated attack protection and the utmost in performance.

The Juniper integrated Stateful inspection firewall and VPN product line will validate session data, protecting against Layer 3 and 4 attacks, and pass large volumes of data through, given that the bandwidth at the central site is most likely of a magnitude higher than that of smaller remote offices. It may also be needed to support hundreds/thousands of VPN connections. The organization will most likely forgo enabling Deep Inspection on the central site firewall, in favor of a stand-alone IDP device to inspect both internal and external traffic. This will enable the organization to take advantage of Juniper IDP's Multi-Method Detection, which goes beyond protocol conformance and attack pattern matching, and advanced logging and investigative capabilities. Maintaining multiple devices does increase network complexity and support costs, but at a central site there should be sufficient IT resources available.

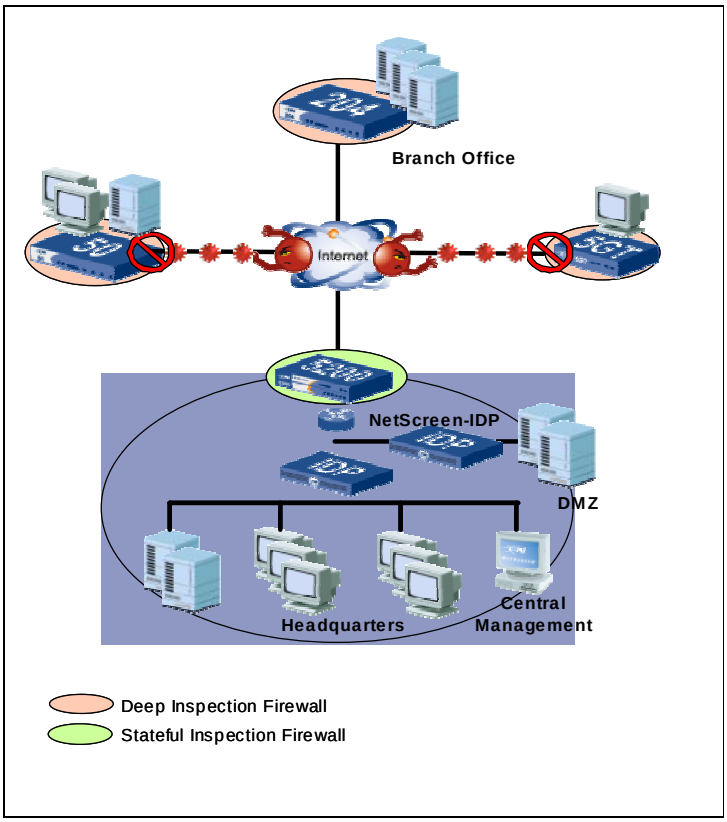
But what happens as you begin to deal with some of the smaller nodes in the enterprise network? Given the distributed nature of today's computing environment, more often than not you will have numerous smaller sites that also require application-level protection. This is where Deep Inspection is perfectly positioned to enhance the organization's security stance. These smaller nodes are at the edge of your network, where there most likely will be far less IT resources available for maintenance and upkeep. Using multiple devices is not a cost-effective strategy, thus by moving application-level protection into the firewall, via Deep Inspection, it is possible to add the necessary protection at these sites. While stand-alone IDP

protects against more than 50 different application layer protocols, Deep Inspection adds application layer protection for those protocols that will typically be used at the remote sites (Web, e-mail, file transfer, DNS). Moreover, if only a subset of these protocols is needed in a particular location, Juniper's DI firewall can be configured appropriately, as described in the previous section. Finally, the automatic upgrade feature of ScreenOS means that IT maintenance costs are brought down even further, making rich application layer protection a reality at the edge of enterprise networks.



The above graphic is a simplified diagram of an entire topology, illustrating the overall range of scenarios that the Juniper product line is designed to protect. At a remote site or telecommuter's office, you could choose configure either a Juniper firewall with Deep Inspection enabled, or just use Juniper's Stateful Inspection, as the situation warrants. Here the integrated capabilities of ScreenOS are truly brought to bear. If for example this site is just an office with possibly just a print server and little or no IT support, it may be sufficient to just use Stateful Inspection. But there are other scenarios where it is prudent to use Deep Inspection. A telecommuter's machine is an unknown quantity in some respects, which bears the question, "how much traffic is work-related and how much is home use?" Does the organization really know what is transpiring on these machines? A security strategy may be to split this network node between the home and work user, and to enable Deep Inspection for the work user to protect critical network resources.

As shown in the below graphic attackers may choose any number of routes to breach the network, so your IT staff needs this integrated capability to adjust your organization's security strategy based on the current security posture.



By layering the Juniper integrated firewall/IPSec VPN systems and appliances and IDP solutions throughout the enterprise network, organizations can maximize their security while minimizing their overhead. Obviously the most critical resources must be protected at all costs, so by layering the firewall functionality in this manner increases an organizations ability to mitigate both network and application-layer attack.

Conclusion

Pervasive application-level protection is critical when securing enterprise networks. The distributed nature of today's computing environment has left the perimeter of the network vulnerable to attack, as the firewalls prevalent today do not protect against application-level attacks, simply because they do not have the means available to do so. With the introduction of Juniper's Deep Inspection firewall, IT administrators have at their disposal an easy-to-manage solution that provides protection to the application-layer protocols that are typically used in the smaller nodes (remote, branch, and regional offices). It is at these locales where Deep Inspection is most appropriate, and the combination of this new firewall technology in conjunction with function-specific devices like stand-alone IDP means that security personnel can rest assured that their entire network has protection against application attacks.

Blank Page