Global Information Assurance Certification Paper

Stateful Inspection
Rainer Kampp
June 17, 2003

Abstract

[4] Stateful Inspection, invented by CheckPoint Software Technologies, is the de facto technology standard for enterprise-class network security solution firewalls. In order to provide robust security, a firewall must track and control the flow of communication passing through it. To reach control decisions for TCP/IP based services (whether to accept, reject, authenticate, encrypt and/or log communication) it is not sufficient to examine packets in isolation. State information, derived from past communications or applications, is an essential factor in making the network secure. Over the past several years, enterprise firewalls have become the staple of network security architectures. Designed primarily to provide access control to network resources, firewalls have been successfully deployed in the large majority of networks. [1] Considering which level of protection and network performance these systems support, following types of firewalls are the most commercial ones used for network security:

- static packet filter
- dynamic (stateful) packet filter
- application level gateway (proxy)
- stateful inspection

Every firewall, whether it is a packet filter firewall, an application gateway proxy-based or stateful inspection firewall, examines and controls the flow of TCP/IP traffic where TCP/IP is the standard for Internet connections and its model is derived from the standard OSI model. Firewalls commonly defeat more than 90% of network attacks. However, while most firewalls provide effective access control, many are not designed to detect and thwart attacks at the application level. The most common vehicle for attacks against the network layer is the Internet Protocol (IP), whose set of service resides within this layer. As with the network layer, the transport layer and its common protocols (TCP and UDP) provide popular access points for attacks on applications and their data.

[7] Today's knowledgeable hackers have advanced well past scanning for open ports on firewalls and are now directly targeting applications. Since application-driven attacks are sophisticated in nature, effective defenses must be equally sophisticated and intelligent. Many firewalls, particularly those based on Stateful Inspection technology, have maintained successful defense arsenals against network assaults. As a result, a growing number of attacks attempt to exploit vulnerabilities in network applications rather than target the firewall directly. This important shift in attack methodology requires that firewalls not only access control and network-level attack protection, but also understand application behaviour to protect against application attacks and hazard.

The OSI- and the TCP/IP- Model

The OSI-Model (Open System Interconnection) describes the communication between computer systems with 7 independent protocol layers:

- Application Layer 7
- Presentation Layer 6
- Session Layer 5
- Transport Layer 4
- Network Layer 3
- Data Link Layer 2
- Physical Layer 1

It is very useful to know these layers because this is essential for understanding all the security measures handled by firewalls.

The Physical Layer describes the electrical specifications of the connection. The Data Link Layer connects the physical part of the network with the NIC (Network Interface Card) of the computer system. Each network card has its own unique physical address, called MAC (media access control) address. The Network Layer is responsible for the routing of the datastream. It also handles the relationship between the MAC address (physical address) and the IP address (logical address). The Transport Layer ensures reliable connectivity from end-to-end. As the TCP protocol is used it also handles the sequencing of packets in a transmission. The Session Layer makes sure that information exchanged across the connection is in synchronization on both sides. The Presentation Layer guarantees that the received format of the data is useful for the system. Sometimes the datastream is compressed because of better throughput across the network. Therefore it must be decompressed at the target system. The Application Layer for example determines if the running application needs network connection and then manages the requests from the running program to the other layers.

The TCP/IP–Model describes the communication between computers in abstraction to the OSI – model with 4 independent protocol layers. Since the TCP/IP communication only uses 4 layers, each of these layers has a little bit more to do. The layers are:

- Application (services like ftp, telnet, smtp, http and others)
- Transport (TCP/UDP) (OSI layer 4 – the transport layer)
- Internet (IP) (OSI layer 3 – the network layer)
- Physical (most used medium is Ethernet)

TCP (the transmission control protocol) is responsible for breaking up the messages into datagrams, putting a header at the front of each datagram, reassemble them at the destination computer, resenting anything that get lost and putting the datagrams in the right order.

TCP is a connection-based protocol; therefore the TCP-Header segment consists of the following fields [8]:

<table>
<tr><td colspan="8">Source Port</td><td colspan="4">Destination Port</td></tr>
<tr><td colspan="12">Sequence Number</td></tr>
<tr><td colspan="12">Acknowledgement Number</td></tr>
<tr><td>Data Offset</td><td colspan="2">Reserved</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F I N</td><td colspan="4">Window</td></tr>
<tr><td colspan="8">Checksum</td><td colspan="4">Urgent Pointer</td></tr>
<tr><td colspan="10">Options</td><td colspan="2">Padding</td></tr>
<tr><td colspan="12">DATA …</td></tr>
</table>

The flags SYN (Synchronize), ACK (Acknowledge) are used to initiate a normal TCP connection. The flag FIN (Finish) is used to finish a connection. For example a web-client initiates the connection to the addressed web-server by sending a SYN packet to the well-known port 80 (http). Port 80 is the port on which by default a web-server is listening. The server then responds with a SYN/ACK packet. The client initiating the connection then finally responds with an ACK packet, and the connection is established. This procedure is called the three-way handshake [8]:

Client	----	SYN	→	Server
Client	←	SYN / ACK	----	Server
Client	----	ACK	→	Server

The connection-based protocol uses sequence and acknowledgement numbers, source and destination port numbers. The port numbers range is from 1 – 65302. Where the numbers lower than 1024 are the well known respective the trusted ports and the higher are the ephemeral ports which means they could be used by any service for any reason. For example FTP-servers are listing on the well-known port 21, HTTP-servers are listening on port 80 and SMTP-servers on port 25. A communication pair between an HTTP-client and an addressed HTTP-server looks like:

HTTP-client:1025 → HTTP-server:80
HTTP-client:1025 ← HTTP-server:80

As long as this connection exists, the port pair 1025:80 for this connection remains the same.

The TCP-Layer sends each of these datagrams to the Internet (IP)-Layer. The IP-Layer does not care about what is in the datagram or even in the TCP header. IP is simply responsible for the routing of the datagrams.

The IP-Header segment has the following fields [8]:

Version	IHL	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source IP-Address			
Destination IP-Address			
TCP Header, DATA …			

The IP-Header contains some additional fields. The flags and fragment offset are used to keep track of the pieces when a datagram has to be slip up. This can happen when datagrams are forwarded through a network for which they are too big. The time to live is a number that is decremented whenever a datagram passes through a system. When it goes to zero, the daragram is discarded and this prevents routing-loopbacks. The type of service describes what protocol is used where the service TCP is indicated with 6, UDP with 17 and ICMP with 1.

UDP (the user datagram protocol) is designed for applications where no sequences of datagrams need to put together. It is a connectionless transport protocol. UDP is used by applications that do not require the level of service of TCP or that wish to use communications services (multicast or broadcast delivery) not available from TCP. As UDP is a connectionless protocol an UDP-Header is shorter than a TCP-header. The UDP-Header segment consists of the following fields [8]:

Source Port	Destination Port
Length	Cecksum
DATA …	

ICMP (the Internet control message protocol) is used for error messages for the TCP/IP software itself and fit into one datagram. This protocol is used for network troubleshooting. Commands like *ping* or *tracert* uses parts of the available message types (such as echo request, echo reply, destination unreachable, time exceeded, etc.) and are very useful for network administrators to monitor the route the packets take to reach the destination IP address.

Packet Filter

[1] The *static packet filter*, historically implemented on routers, examine each and every packet at the network layer – OSI layer 3 - and compare them to the configured access lists. The administrator can define rules that determine which packets are accepted and which packets are denied. These rules are called the security policy.

The *dynamic (stateful) packet* filter is an advanced packet filter that operates up into the transport layer – OSI layer 4 – to collect additional state information. In simplest terms, the typical dynamic packet filter is aware of the difference between a new and an established connection. Once a connection is established, it is entered into a table. All packets are compared to this table. When the packet is found to be an existing connection, it is allowed to pass without any further inspection. The state awareness provides measurable performance benefit.

Because the packet filtering operates only at network layer it has low impact in network performance and is low in cost. Thus only the IP and TCP headers are examined it only provides a low level of protection. It cannot know the difference between a real and a forged address. If an address is present and meets the packet filter rules along with the other rule criteria, the packet will be allowed to pass.

Suppose the administrator took precaution to create a rule that instructed the packet filter to drop any incoming packets with unknown source address. This packet-filtering rule would make it more difficult, but not impossible for a hacker to access at least some trusted servers with IP addresses. The attacker could simply substitute the actual source address on a malicious packet with the source address of a known trusted client. This common form of attack is called *IP address spoofing* and is very effective against a packet filter. Although the performance of a packet filter can be attractive, this architecture alone is generally not secure enough to keep out hackers determined to gain access to the protected network.

The static packet filter is not state aware. The administrator must configure rules for both sides of the conversation to a protected server. For many services such as FTP and SMTP an administrator of a static packet filtering firewall has to open an entire range of ports with static packet filtering rules.

Also import is that many vendors of packet-filtering firewalls fail to follow RFC recommendations in the establishment of the connection and opens a connection without the three-way handshake. The attacker can simply spoof the trusted host address and fire any of the many well-known single packet attacks like *Ping of Death and TearDrop* at the firewall and or servers protected by the firewall while maintaining his complete anonymity.

Application Level Gateway (Proxy)

[1] An application level gateway intercepts incoming and outgoing packets, runs proxy servers that copy and forward information across the gateway preventing any direct connection between a trusted server or client and an untrusted host. The proxies are application specific (such as http, smtp, ftp, etc.) and examine the entire packet and can filter at the application layer of the OSI model.

For example only an HTTP proxy can copy, forward and filter HTTP traffic. In other words, a connection from a host to a server is actually opened to the proxy. If the proxy determines that the connection is allowed, it opens a second connection to the

server itself on behalf of the original host. The data portion of each packet must be stripped off, examined and rebuilt and sent again on the second connection. This examination and handling of packets through a second connection means that proxy firewalls are very slow. Proxies are limited as firewalls, because they must understand the application layer. As new protocols are developed, new proxies must be written and implemented to handle them. Therefore, proxies only support a handful of the more common protocols.

If a network relies only on an application level gateway, incoming and outgoing packets cannot access services for which there is no proxy. If an application level gateway ran FTP and HTTP proxies, only packets generated by these services could pass through the firewall. All other services would be blocked. Strong application proxy that inspects header length can eliminate an entire class of buffer overrun attacks. But these proxies must be written securely. Historically some vendors have introduced buffer overruns within the application gateway itself.

With proxy firewalls, you must establish a TCP session with the firewall itself if you want to access a service on the other side of the firewall. A proxy of application being accessed then inspects the data that is transmitted. The dual advantage here is that you have a centralized location from which to deal with TCP level attacks, and one point from which to ensure that a hacker is not trying to exploit any vulnerabilities that may be associated with this application. If the proxy application detects no problems, the firewall establishes another connection with the destination device. This is the primary advantage of application level gateway firewalls because no direct connections are allowed through the firewall. The disadvantage is that this firewall is not transparent for the internal hosts, which want to connect to an external server. Each internal client host must be configured to be aware of the firewall and must have client software that is designed to be capable of communicating with the proxy software on the firewall. In modern secure environments the client do not either have to be aware of the firewall or run special software to communicate with the external network.

The reason for the difference in speed of packet filters and application gateways is a function of the amount of security provided by the firewall. [3] With current hardware platforms only connections requiring more than 75-100 Mbps throughput per gateway must consider packet filter firewalls. Since most organizations use a maximum of 2 Mbps as Internet connections, only Intranet applications on the internal high-speed network are forced to seriously consider packet filters. Application gateways are capable of supporting the common applications in use on the Internet by providing the highest level of protection.

Stateful Inspection

[1] Stateful inspection combines the many aspects of dynamic packet filtering and application level gateways (proxies). While stateful inspection has the inherent ability to inspect all seven layers of the OSI model, most installations only operate as a dynamic packet filter at the network layer because of the dramatic impact of perfor-mance.

Stateful inspection firewalls keep state information about connections. They understand that a single connection between two computers generally consists of many packets, and that they only need to compare the first packets of a given connection against the defined security policy. Once a connection has been established, it is recorded in a table. The table which includes the connection information such as IP addresses, TCP ports, sequence numbers is checked for each packet that arrives at the firewall and if the packet belongs to an existing connection it is allowed to pass. Since the security policy is consulted once for each connection, complex security polices do not greatly impact performance.

Like an application level gateway, stateful inspection can be configured to drop packets that contain specific commands within the application header. For example, the administrator could configure a stateful inspection firewall to drop HTTP packets containing a "put" command. This certainly reduces the performance of the firewall system and many administrators operate the stateful inspection based firewall as nothing more than a dynamic packet filter. Unlike an application level gateway, stateful inspection does not break the client-server model to analyze application layer data. They do not rewrite packets. Stateful inspection relies on algorithms within an inspect engine to recognize and process application-layer data. These algorithms compare packets against known bit patterns of authorized packets and respective vendors have claimed that theoretically they are able to filter packets more efficiently than application-specific proxies are. But the result is that this higher throughput reduces security. [11] Most stateful inspection firewalls employ a keyword-like filtering methodology. They typically filter for bad keywords in the application payload. There will always be new bad things created by malicious users and must be updated by incident.

[2] UDP and ICMP do not themselves contain any connection information (such as sequence numbers). However, at the very minimum, they contain an IP address pair. UDP also contains port pairs, and ICMP has type and code information. All of these data can be analyzed in order to build "virtual connections" in the cache. For instance, a cache entry will be created by any UDP packet, which originates from the internal network. Its IP address and port pairs will be stored. For a short period of time, UDP packets from the Internet which have matching IP and UDP information will be allowed back in through the firewall. An analogous situation exists for ICMP. Only outgoing echoes will allow incoming replies, and outgoing timestamp requests will allow incoming timestamp replies. No other ICMP packets are allowed in through the firewall, simply because they are too dangerous and contain too little tracking information.

Some higher layer protocols (such as ftp) uses multiple network connections simultaneously. They usually have a "control connection" (ftp server listening on well-known port 21) which is used for sending commands between the two computer systems, and then "data connections" (ftp-data server sending on well know port 20) which are used for transmitting information. Consider the FTP protocol. A user on the internal network opens a control connection to a server on the Internet and requests a file. At this point, the remote server will open a data connection from the Internet. For FTP to work properly, this connection must be allowed to pass through even so a connection from the Internet would normally be dropped. In order to achieve this, a

stateful inspection firewall inspects the application-level FTP data. It searches for outgoing "port" commands, and when it sees these, it adds a cache entry for the anticipated data connection. To track and act on both state and context information for an application is to treat that traffic stateful. Tracking the actual state of each communication session enables the firewall, among other things, dynamically open only those ports that are required by an authorized session.

True Stateful Inspection

[6] To track context, a firewall must inspect the content of packet payloads to ensure that each packet entering the networks meets the expected parameters and attributes of the communication session. This guarantees that malicious packets that do not fit the context of the communication cannot circumvent firewall security. The following are examples of state and context-related information that a firewall should track and analyze:

- Packet header information (source address, destination address, type of service, source port, destination port, packet length)
- Connection state information (which ports are being opened for which connection)
- Sequence and acknowledgement numbers, fragment offset
- Packet reassemble, application type, context verification (i.e. the packets belongs to the communication session)
- Incoming and outgoing firewall interface
- Date and time

True Stateful Inspection means tracking the state and context of all communications. But there are firewalls where the implementation of Stateful Inspection is by default incomplete because of the higher packet throughput this system can handle by reducing the stateful ability. These systems can by set to the "fast path" mode that means the firewall only inspects packets in which the SYN flag is set. If an attack is fragmented into multiple packets these firewall is unable to detect and block the attack. Given how easy it is to fragment packets, this weakness renders networks protected by this firewall vulnerable to a broad range of well-known attacks. [9] For example the *TearDrop* attack exploits a vulnerability in the assembling of data packets sent over the network. When you send a big file over the Internet, it is broken down into several small packets of data. These are sent to a server with an offset field. In the *TearDrop* attack the offset fields are modified, so that they overlap and confuse the server. The server does not reassemble the file until it gets it last packet, so what the attacker does is send a continuous stream of overlapping packets. The queue fills up and begins to reduce the memory of the computer and the machine either freezes or slows down.

[7] Any traffic not adhering to strict protocol or application standard must be closely analyzed before it is permitted into the network; otherwise business-critical

applications may be put at risk. For example binary data in HTTP headers are prohibited by the official HTTP standard. Thus most of the firewalls do not check this as a result many hackers launch attacks by including executable code in HTTP headers. Also the HTTP standard does not limit header length, excessive length should be blocked or flagged to reduce the chance of buffer overflows and to limit the size of code that can be inserted using the overflow. Malicious data can also enter the internal network by embedding itself in URLs. An application such as an email client could automatically execute an HTML-embedded URL. If the URL was malicious, damage to the network or the user's system may occur. Therefore access to potentially malicious URLs should be blocked or limited.

Not only application-layer communications introduce malicious data to a network; the application itself might perform unauthorized operations. A network security solution must have the ability to identify and control such operations. A firewall should place connection restrictions on particular file names and controls potentially hazardous FTP commands like PUT, GET, SITE and REST. For example, a security policy may require operational restrictions on all files containing the word "payroll".

Preventing malicious manipulation of network-layer protocols (e.g. ICMP) is a crucial requirement for multi-level security firewalls. ICMP allows one network node to ping or send an echo request to other network nodes to determine their operational status. This capability can be used to start a "smurf" DoS (Denial of Service) attack. The smurf attack is possible because standard ICMP does not match requests to replies. Therefore an attacker can send a ping with spoofed source IP address to an IP broadcast address. The IP broadcast address reaches all IP addresses in a given network. All machines within the pinged network send echo replies to the spoofed and innocent source IP. Too many pings and responses can flood the spoofed network and deny access to legitimate traffic. Dropping replies that do not match requests can block this type of attack. Stateful inspection handles this attack by creating virtual session information for connectionless protocols (such as UDP and ICMP).

Another network-layer event is the *PortScan*. A port scan does what the name implies: a hacker scans a range of ports on a target host in hopes of identifying and exploiting weaknesses on running applications. The reconnaissance that a port scan performs is a hazard then can lead to an attack. A security gateway must be able to raise alerts and block or shutdown communications from source of the scan.

At least a firewall must defend against variations of well-known attacks (e.g. Code Red or Nimda). There are firewall devices where the attack patterns (e.g. malicious URL) must match identically to those in the firewall database. Any variation of an attack, no matter how trivial, will traverse the firewall undetected. This is a direct result of the fact that some firewalls does not support regular expression matching, which gives the administrators the ability to look for attack variants using wild card definitions.

Performance and Security Considerations

Performance (packet throughput and simultaneous connections) and security on the other side are the both aspects that you must consider when you use a firewall as a security measure. [3] The security expert Bill Stout wrote on the firewall mailing list: "The purpose of a security device is to protect a network, not to be fast. Fast is what airline travellers want when passing through airport security, secure is what they want when they tumble through the air after their plane blows up."

As we learned the highest level of protection is achieved by the Application Level Gateway (Proxy) firewall because all the 7 OSI layers were inspected and analyzed. The Stateful Inspection firewall also looks up to the 7 layers but not so deep in content. It provides "light proxies", that do not intercept the client/server communication. Therefore this firewalls should be faster than Proxy firewalls but cannot support the same protection level. The packet filter firewall even the dynamic packet filter has the highest packet throughput but the less security protection, because only the OSI layer 3 respective layer 4 is examined.

The performance of these firewalls is dependent on what processors are used and if the systems support symmetric multi-processing architecture (SMP). This technology has considerable reduced the performance gap between the packet filtering firewalls and the application gateway proxy-based firewalls.

Despite the continuous improvement in network security technologies, it is no longer enough to have a single line of defense. Defense in depth is the practise of layering two or more firewalls to increase data protection. Important for this measure is that you use firewalls with different architectural designs (Packet Filter Firewalls, Stateful Inspection Firewalls and Proxy-based Firewalls) from different vendors. So when a hacker compromised one firewall, it will take him a lot of longer to compromise the other(s). The point of this is to slow down the time it takes to compromise a system beyond this multiple barriers.

Conclusion

Stateful packet filters are faster than application level gateways. They have a better performance but less security. To achieve the highest level of protection in combination with the highest network performance you can use both technologies. The application level gateway (proxy) as external firewall to the Internet, to achieve the highest security by an adequate throughput and a True Stateful Inspection packet filter as internal firewall to the Intranet, where high TCP/IP traffic throughput is more necessary and important.

References

[1] Paul Henry, “An Examination of Firewall Architectures”, April 2001
http://www.cyberguard.com/news_room/whitepapers.cfm

[2] SSI Service Strategies Inc., “Stateful Inspection”, March 2003
http://www.ssimail.com/Stateful.htm

[3] Trusted Information System Inc., “Application Gateways and Stateful Inspection”, January 1998
http://www.spirit.com/CSI/Papers/apgw+spf.html

[4] Sofaware, “Stateful Inspection”, 1999
http://www.sofaware.com/html/tech_stateful.shtm

[5] Stonesoft, “Multilayer Inspection”, February 2003
http://www.stonesoft.com/files/products/StoneGate/SG_Multi-layer_Inspection_Whitepaper.pdf

[6] Check Point, “Check Point on Security”, 2002
http://www.sapphire.net/docs/stateful%20inspection%20comp%20white%20paper.pdf

[7] Check Point, “Check Point Application Intelligence”, 2003
http://www.checkpoint.com/products/downloads/application_intelligence_whitepaper.pdf

[8] Charles L. Hedrick, “Introduction to the Internet Protocols”, 1987
http://oac3.hsc.uth.tmc.edu/staff/snewton/tcp-tutorial/sec2.html

[9] “Denial of Service Attacks”, June 2002
http://www.symonds.net/~deep/stuff/tekmail/issue35.php

[10] Check Point, “VPN-1/FireWall-1 Management I”, 2001
Student Guide Check Point 2000 Edition

[11] Paul Henry, “Protocol and Application Awareness”, 2003
http://www.cyberguard.com/pdf/cyberguard_whitepaper_protocol_application_awareness.pdf