Modes of firewall operation

Different Types of Firewall

Firewalls operate in different ways, depending on the layer on which they are implemented.

Circuit level firewall: data link layer

Various names are used for this type of firewall. It is the type provided whenever NAT and PAT technology are used (see 3.7).

When a protected computer starts a conversation with a remote computer, the traffic is intercepted by the circuit level firewall, which forwards the request. When the return traffic reaches the firewall, the internal tables are checked to establish if it needs forwarding to a protected computer or if it is a non-requested conversation.

The key advantage of this kind of firewall is that only return traffic from conversations that were initiated from behind the firewall will be allowed through. As there is no direct connectivity between the protected computer and the external network, any unrecognised conversations are dropped. However, this can also be a disadvantage as anything requested by the protected computer will be received even if it is malicious content. SOHO routers, commonly used for broadband connections at home, generally provide a circuit level firewall through NAT and/or PAT.

Packet filtering firewall: network layer

Firewalls acting at the network layer were the first to be developed and are probably the best understood by network administrators. They work by examining each packet against a set of defined rules.

These rules usually relate to:

- source and destination IP addresses
- source and destination ports
- protocol at transport or network layer (IP, TCP, UDP, ICMP etc)
- physical interface
- direction (ingress or egress)
- packet state.

On the whole, packet filters can only allow or drop and log traffic. The traffic contents are not altered. When a packet is received it is evaluated against a set of rules and once the packet matches a rule the defined action is taken. This means that the order of the rules is critical as

the first match found determines what happens to that particular packet. Rules are easily defined using simple logic. For instance, to block all incoming SMTP traffic, a rule can be defined for all TCP source traffic to the local destination network matching SMTP port 25.

Packet filtering can also remove other network traffic. For example, it can match specifically TCP or UDP as well as ICMP at the network layer or IGMP.

Application firewall: application layer

Firewalls acting at the application layer inspect traffic at a much higher level than traditional firewalls. They can be network devices placed inline, proxy servers to handle specific traffic or applications running on a server to filter traffic to a particular program.

Firewalls on the application layer operate differently to those on the network layer because of how data is transmitted across networks. Each chunk of data consists of two parts, the 'header' and the 'payload'. (Using a postal analogy, the header is the envelope and the payload the letter inside.) Conversations between computers comprise many of these chunks of data, known as packets. An application layer firewall can inspect the payload as well as the header and can look at a series of packets together.

One of the key features of an application layer firewall is its facility to block any packets that do not comply with the RFC standard for the protocol being inspected. For example, an exploit against a web server that uses a subtly altered HTTP packet will be blocked. An application layer firewall can also act as a content filter: by examining the payload, packets containing Java™, ActiveX® or malware can be blocked. Content can even be reassembled and virus checked before being passed on to the end user.

In most modes, the operation of such a firewall is transparent to the user, except for the time lag caused by the decoding of the complete packet. With faster performance devices and the optimising of rule sets this is now rarely a concern, though earlier dedicated proxy servers required configuration of individual applications.

A relatively new term for the unwanted content that can be filtered by an application layer firewall is Anti–X. This covers a range of different areas, including anti-virus, ad/spy/ malware, worms, spam and phishing.

Traditional proxy server/web cache appliances can sometimes offer some of the features of an application firewall as well as the advantages of caching contents. However, most application layer firewalls are also excellent reporting tools and can generate numerous auditing logs in combination with facilities for authentication and real-time alerting. The term 'application firewall' can also be used for applications which intercept content for sanity checking before passing it to the ultimate destination. URL Scan is an example of such an application, which filters the URLs passed to a Microsoft® IIS system.

Firewall Settings

Default allow and default deny

There are two different types of firewall policy: default allow and default deny. The default allow firewall rule set allows all connections through the firewall unless otherwise stated.

There is no implicit or explicit ‘deny all’ at the end of the rule set.

Early firewall policies just blocked a few known malicious signatures or activities, but as the threat level and frequency of attacks increased over time, it was recognised that it was more prudent to implement a default deny policy.

In addition, with some default allow firewalls there is a short time lag between when the firewall code is initialised and when additional rules are loaded. This presents a security risk for that short window of time.

A default deny firewall rule set will deny all connections through the firewall unless a connection matches a specific rule. An empty default deny rule set has no effective connectivity. Many current firewall systems are configured as default deny and do not even require an explicit ‘deny all’ statement at the end of the rule set. However, for the ease of logging and debugging, it is best to add the statement explicitly as a reminder.

For example, when monitoring an access list on a Cisco device, without explicit deny rules it is difficult to get a breakdown of the dropped packets using the command:

Janetrouter> sh access-list 101

Separate deny statements will provide further statistics on each protocol being dropped:

access-list 101 deny tcp any any log

access-list 101 deny udp any any log

access-list 101 deny ip any any log

A firewall cannot possibly detect or prevent all possible threats. However, a default deny configuration will provide limited protection against zero day attacks using new port ranges, instead of none at all.

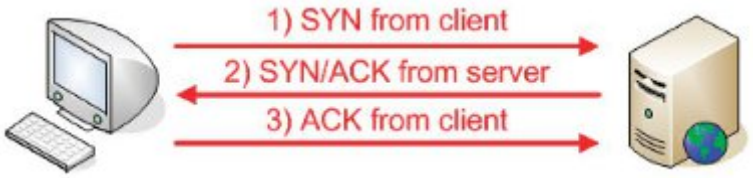
Stateless and stateful

Firewalls were initally stateless, examining each packet individually against the firewall rules. The firewall has no information as to whether the packet is the start of a new connection or part of an established one or any memory of packets that may or may not have gone before.

A stateful firewall can identify conversations and track activity to deny new connections from a hostile network, while permitting established connections to traverse the firewall. This is achieved through an internal table of attributes for each connection: IP addresses, ports, direction of flow and in some cases sequence numbers. Subsequent packets, if matched in the internal table, are then forwarded with less stringent examination because they are part of an established connection.

In the case of a three way handshake, a stateless firewall would examine each embryonic packet that forms part of the handshake. A stateful firewall would recognise the three way handshake, the continuing established connection and the resultant tear down at the end. The first packet sent from the client with the SYN bit set is recognised as part of a new connection by the firewall. The server will reply with a packet where the SYN and the ACK bit is set; this

half-open state is allowed back through the firewall and recorded in the internal table. The final stage of the client replying with a packet with the ACK flag set brings up the TCP connection in the established state. UDP does not use the same three way handshake as it is a connectionless protocol, so a firewall will deem a UDP connection to be established at the receipt of the first packet.



[1]

Figure 3: Flags in TCP three way handshake

The key advantage to a state table is that the rule set is only interrogated when the first packet with the SYN bit set is received by the firewall or when the rule set changes. The subsequent packets making up the same connection will be passed or blocked without rechecking the whole rule set. When the firewall rule set changes the state table must be cleared and all initial connections are again matched against the new rule set. Sessions in the state table will time out when no traffic has been received for a period of time or when there is a flood of packets with the SYN bit set from the same host which reach a pre-defined threshold.

Level of inspection

DPI Layer 7 inspection and other similar terms refer to the level at which a firewall, or indeed any network device examines packets.

The technology used by DPI firewalls is the same as in application firewalls and the two are often combined in modern firewalls. Both the ‘header’ and the ‘payload’ of packets will be inspected and with this extra data the firewall will be able to match packets against more complex rules.

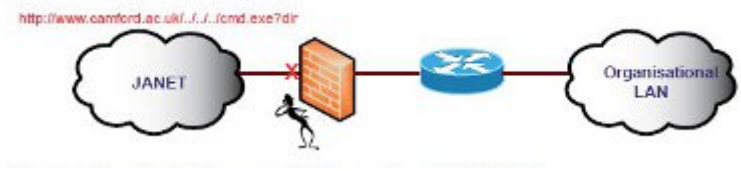
A DPI firewall will use a set of signatures, more akin to an IDS or IPS. When a packet is matched against a signature, it can be dropped, tagged (including QoS marked), rate limited and logged. Packet signatures can be simple HTTP requests with **cmd.exe** in the URL or more complex indicators of NetBIOS/SMB worm activity.

DPI systems have a considerable overhead, which is why manufacturers are starting to implement their code in silicon ASICs to obtain faster wirespeed performance. This allows the device to maintain the state table for the stateful firewall as well as the current state of the application using the conversation.

Placing the detection technology on the firewall allows malicious packets to be detected and dropped earlier in their ingress to the network. Traditionally, an IDS system would be able to alert the network administrator to such activity, or an IPS system would use source quench, send resets or activate rules using tools like **shun**. Moving this technology closer to the malicious network can only improve the security of an organisation.

The requirement for IDS and IPS systems is not diminished, however, as it is impractical for a

firewall to monitor the number of rules that a dedicated IDS/IPS system can at a backbone wirespeed. Moreover, additional functionality enhances the security in depth principle.



[2]

Figure 4: Deep Packet Inspection blocking malicious HTTP URLs

Hardware Firewalls

Hardware firewalls have long been considered a superior firewall platform. The main reasons for this are speed related: hardware manufacturers have speeded up packet throughput using hardware ASICs and optimised code. However, with advances in modern hardware platforms and the additional DPI features of newer firewalls, the speed differential is narrowing.

Cisco offers the PIX hardware solution, a packet filtering firewall with stateful inspection. There are several DPI features in newer versions which enhance the rules that can be created. Some PIX functionality has recently been introduced in the Cisco ASA.

There are many other hardware firewall vendors, including SecureIron® perimeter traffic manager from Foundry Networks® and the SSG and NetScreen products from Juniper®.

Firewall Software

Off the shelf software

There are many software packages available to provide firewall functionality on a range of platforms. There are a number of factors to be assessed in chosing a software solution but it is important to remember the firewall is only as secure as the underlying operating system and its configuration.

In addition to purchasing the software, hardware will need to be specified appropriately. Consideration should be given to CPU and memory throughput as well as the performance of the bus connecting the network adaptors. The yearly maintenance cost for the software needs to be factored into the recurring budget, along with hardware platform maintenance and support.

The Microsoft® ISA server provides web proxy functionality, VPN endpoint termination and a stateful DPI firewall. The latest version of Microsoft® ISA server was released in 2006.

Microsoft® ISA server integrates well with AD and many of its rules can be associated with AD users and computers to provide a better match for security policies. Network load balancing is included along with the ability to publish web pages securely to IIS servers within the organisational DMZ. It is also easier to facilitate firewall rules for services like Outlook® Web Access.

There are two versions of ISA Server, standard and enterprise. The enterprise version supports clustering and load balancing across multiple ISA Servers and requires Windows

Server® 2003.

The Check Point Firewall-1® solution is a CD-based install onto Windows®, Sun Solaris™, Red Hat®, Nokia's bespoke hardware platform or a bare metal customised Check Point SecurePlatform™. It provides a stateful firewall with DPI, a host of additional features and extensive management, status, reporting and auditing capabilities for managing one or many devices.

The newer NGX™ platform offers more granular multicast control as well as enhancing the number of network-based applications supported without having to write extensive rule sets in-house.

Open source

There are a range of open source UNIX/Linux-related firewall systems, all of which run on an installed operating system. However, this software suffers in the same way as other software in the area of wirespeed performance.

NetFilter is the firewalling code in the Linux kernel and provides a packet filter firewall, NAT, Connection Tracking and other features. IPTables is the tool which creates the rules to be managed by NetFilter. IPTables code replaced IPChains from Linux 2.2 (which in turn replaced the ipfwadm code in Linux 2.0). IPTables is far superior to IPChains as it allows the firewall code to operate in a stateful manner monitoring the state of the connection using the tracking layer of NetFilter.

Rules are grouped into chains, which are sets usually following a common thread, for example, protocol or netblock. Instead of a packet being checked against each rule in a long list, the check can branch to different chains to obtain a closer match and reduce the time lag. If a check reaches the end of the chain without finding a match, then the global policy for that chain dictates the action, usually to drop the packet.

IPFirewall (or ipfw) is a FreeBSD® and Mac OS X-based IP packet filter which has the additional functionality of a traffic accounting feature. IPFilter (or ipf) is installed by default in Solaris 10, FreeBSD and NetBSD. Both ipfw and ipf are available as loadable kernel modules or can be included in a fresh kernel.

IPCop is a Linux-based distribution designed purely to provide a firewall platform. The firewall is based on the Linux NetFilter code and provides the same stateful firewall in an easy to manage solution.

There are a number of add-ons for IPCop which provide Anti-X style functionality along with traffic data reporting and the SNORT IDS.

M0n0wall is an embedded firewall distribution based on FreeBSD. It can be installed like IPCop, or can run from a LiveCD or on an embedded system. M0n0wall provides a stateful packet filter firewall, NAT, VPN endpoint termination and a captive portal.

Links
[1] http://community.ja.net/system/files/images/firewalls-tg-03.jpg

[2] http://community.ja.net/system/files/images/firewalls-tg-04.jpg