4.2 Types of Firewalls /DKo98/

FIREWALL CHARACTERISTICS

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible.

2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies.

3. The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

There are three main classes of firewalls: packet filters, application and circuit gateways (proxies), and stateful inspection (or smart filter) firewalls.

4.2.1 PACKET FILTERS

Packet filtering firewalls were the first generation of firewalls. Packet filters track the source and destination address of IP packets permitting packets to pass through the firewall based on rules that the network manager has set (see Figure 4.2.1).

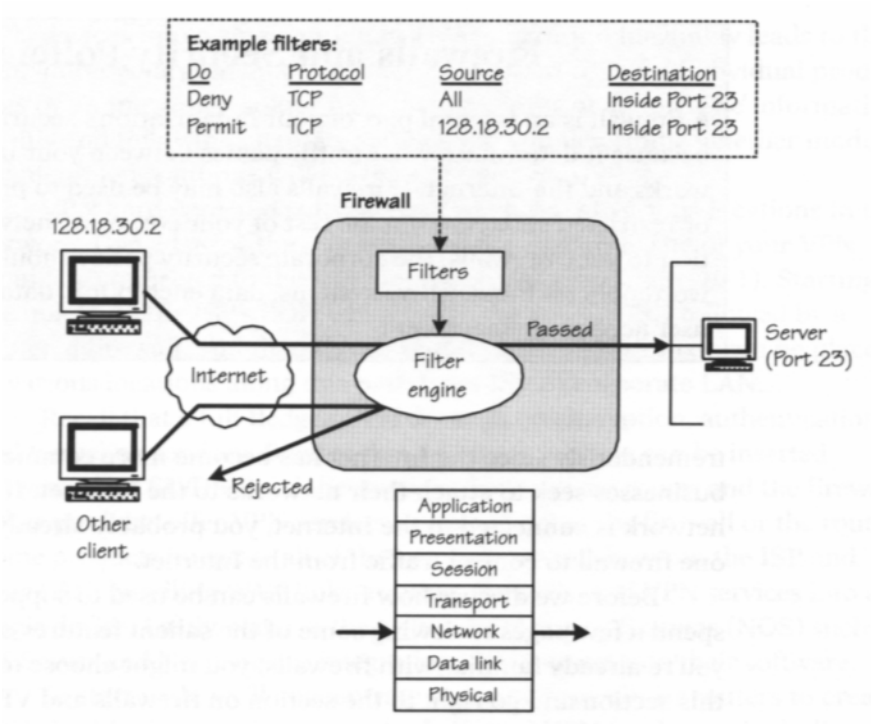

Fig. 4.2.1: Packet Filtering

Two advantages of packet filter firewalls are that they are fairly easy to implement and they're transparent to the end users, unlike some of the other firewall methods we'll discuss shortly. However, even though packet filters can be easy to implement, they can prove difficult to configure properly, particularly if a large number of rules have to be generated to handle a wide variety of application traffic and users.

Packet filtering often doesn't require a separate firewall because it's often included in most TCP/IP routers at no extra charge. Of course, if you're planning to use packet filtering in a router as part of your security policy, you should ensure that the router itself is secure.

But, packet filtering is not the best firewall security you can get. One of its deficiencies is that filters are based on IP addresses, not authenticated user identification. Packet filtering also provides little defense against man-in-the-middle attacks and no defense against forged IP addresses. Also, packet filtering depends on IP port numbers, which isn't always a reliable indicator of the application in use; protocols like Network File System (NFS) use varying port numbers, making it difficult to create static filtering rules to handle their traffic.

Packet filters can be used as a part of your VPN, because they can limit the traffic that passes through a tunnel to another network, based on the protocol and direction of traffic. For example, you could configure a packet filter firewall to disallow FTP traffic between two networks while allowing HTTP and SMTP traffic between the two, further refining the granularity of your control on protected traffic between sites.

4.2.2 APPLICATION AND CIRCUIT PROXIES

Since they're based on address information, packet filters look exclusively at some of the lower layers of the OSI model. Better, more secure firewalls can be designed if they examine all layers of the OSI model simultaneously. This principle led to the creation of the second generation of fire-walls: application and circuit proxies. These firewalls enable users to utilize a proxy to communicate with secure Systems, hiding valuable data and Servers from potential attackers.

The proxy accepts a connection from the other side and, if the connection is permitted, makes a second connection to the destination host on the other side. The client attempting the connection is never directiy connected to the destination. Because proxies can act on different types of traffic or packets from different applications, a proxy firewall (or proxy Server, as it's often called) is usually designed to use proxy agents, in which an agent is programmed to handle one specific type of transfer, say FTP traffic or TCP traffic. The more types of traffic that you want to pass through the proxy, the more proxy agents need to be loaded and running on the machine.

Circuit proxies focus on the TCP/IP layers, using the network IP connection as a proxy (see Figure 4.2.2). Circuit proxies are more secure than packet filters because computers on the external network never gain information about internal network IP addresses or ports. A circuit proxy is typically installed between your network router and the Internet, communicating with the Internet on behalf of your network. Real network addresses can be hidden because only the address of the proxy is transmitted on the Internet.

Circuit proxies do not examine application data; application proxies, which we'll get to next, do that. When a circuit proxy establishes a circuit between a user and the destination, the proxy doesn't inspect the traffic going through the circuit, which can make the proxy more efficient than an application proxy, but may compromise security.

On the other hand, circuit proxies are slower than packet filters because they must reconstruct the IP header to each packet to its correct destination. Also, circuit proxies are not transparent to the end user, because they require modified client software.

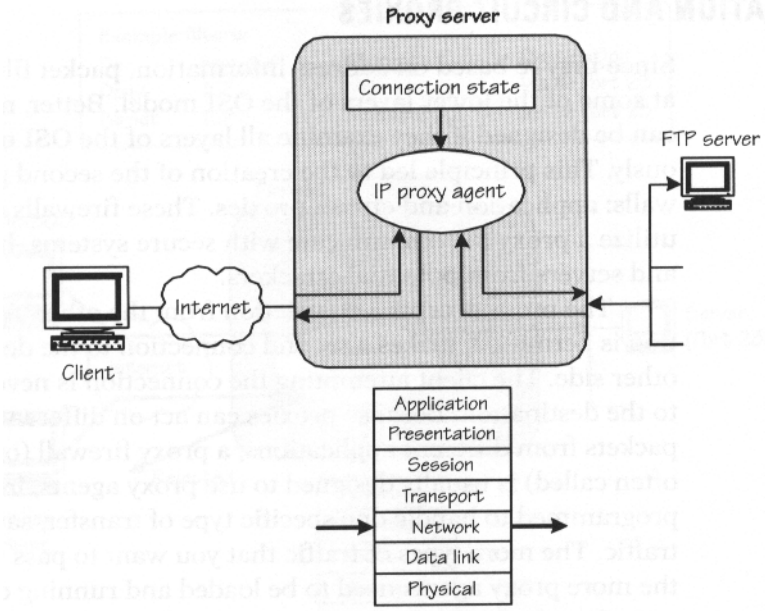

Fig. 4.2.2: Circuit Proxy

Application proxies examine the actual application data being transmitted in an IP packet (see Figure 4.2.3). This approach thwarts any attackers who spoof IP packets to gain unauthorized access to the protected network. Because application proxies function at the Application layer of the OSI model, they also can be used to validate other security keys, induding user passwords and service requests.

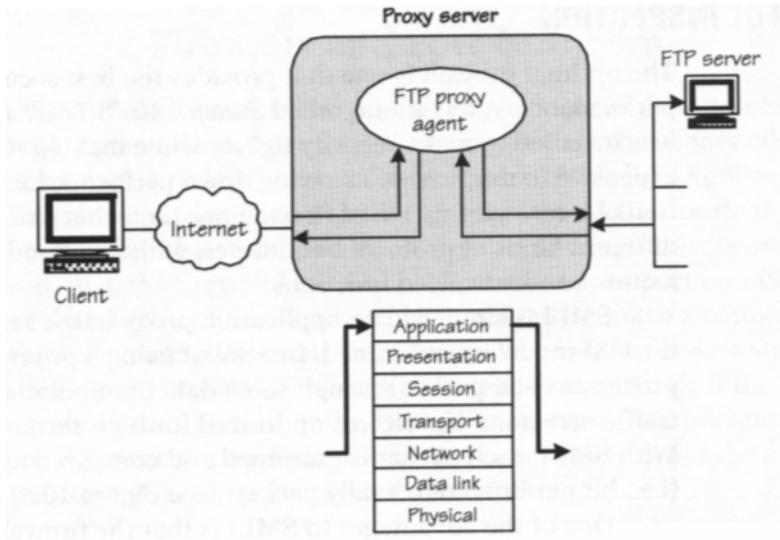

Fig. 4.2.3 : Application Proxy

Proxy firewalls often require two copies of an agent running for each service: one copy to communicate with the internal hosts and one to communicate with the external hosts. Thus, an application proxy may have two copies each of FTP, HTTP, and telnet agents. A circuit proxy operates in a similar fashion; it may have one copy of TCP for the internal network and one copy for the external network.

Because application proxies operate as one-to-one proxies for a specific application, you have to install a proxy agent for every IP Service (HTTP/HTML, FTP, SMTP, and so on) to which you want to control access. This leads to two of the disadvantages of application proxies: a lag usually

exists between the introduction of new IP services and the availability of appropriate proxy agents; and the application proxy requires more processing of the packets, leading to lower performance. Furthermore, many of the application proxies require modified client software, although the firewall's operation is becoming transparent to end users in many of the newer application proxy firewalls.

One important differentiating feature of application proxies is their capability to identify users and applications. This identification can enable more secure user authentication, because digital certificates or other secure token-based methods can be used for identifying and authenticating use

SOCKS

Since circuit-level proxies can offer adequate security for many networks and because some users don't want to pay the price of lower performance found in application-level proxies, a standard for circuit proxies, called SOCKS, has been developed. The SOCKS proxy is designed to pass only SOCKS-related traffic, so SOCKS client software has to process all traffic being passed to the proxy for the traffic to be recognized. No other proxies are included in a SOCKS proxy firewall.

SOCKS was designed for TCP-based client/server applications, using a proxy data channel to communicate between the client and the server. In a SOCKS environment, an application client makes a request to SOCKS to communicate with the application server. This request includes the application server address and the user's ID. SOCKS then establishes a proxy circuit to the application server and relays information between client and server. With SOCKS version 5, authentication and support for UDP relay have been added. SOCKS is commonly implemented as a circuit-level proxy that has enhanced features, such as auditing and alarm notifications, so that it offers many of the features expected of a firewall.

The downside to SOCKS is that client applications must be specially coded for SOCKS or application-level proxies.

4.2.3 STATEFUL INSPECTION

The optimal firewall is one that provides the best security with the fastes performance. A technique called Stateful Multi-Layer Inspection (SMLI) was invented to make security tighter while making it easier and less expensive to use, without slowing down performance. SMLI is the foundation of a new generation of firewall products that can be applied across different kinds of protocol boundaries, with an abundance of easy-to-use features and advanced functions.

SMLI is similar to an application proxy in the sense that all levels of the OSI model are examined. Instead of using a proxy, which reads and processes each packet through some data manipulation logic, SMLI use traffic-screening algorithms optimized for high-throughput data parsing.
With SMLI, each packet is examined and compared against known state (i.e., bit patterns) of friendly packets (see Figure 4.2.4).

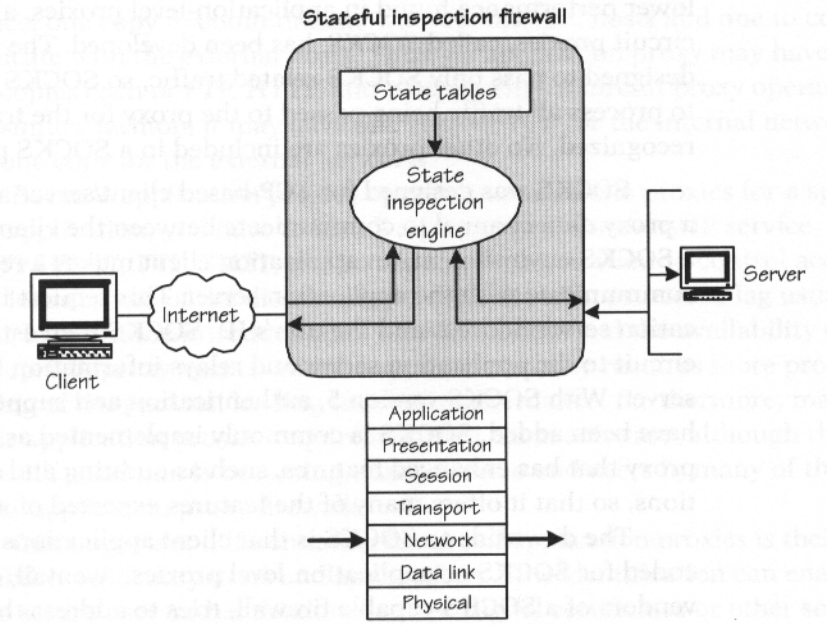

Fig. 4.2.4 : Stateful Inspection

One of the advantages to SMLI is that the firewall closes all TCP ports and then dynamically opens ports when connections require them. This feature allows management of services that use port numbers greater than 1,023, such as PPTP, which can require added configuration changes in other types of firewalls. Statefu inspection firewalls also provide features such as TCP sequence-number randomization and UDP filtering.

What firewalls inspect is depicted in Fig. 4.2.5.

Packet Filtering

Data link header	Internet header	Transport header	Application header	Data

Circuit Filtering

Data link header	Internet header	Transport header	Application header	Data	+ Connection state

Application Gateway

Data link header	Internet header	Transport header	Application header	Data	+ Connection state & application state

Fig. 4.2.5: What firewalls inspect