9.3 Firewalls

9.3.1 Introduction

The dramatic rise and progress of the Internet has opened possibilities that no one would have thought of. We can connect any computer in the world to any other computer, no matter how far the two are located from each other. This is undoubtedly a great advantage for individuals and corporate as well. However, this can be a nightmare for network support staff, which is left with a very difficult job of trying to protect the corporate networks from a variety of attacks. At a broad level, there are two kinds of attacks:

- Most corporations have large amounts of valuable and confidential data in their networks. Leaking of this critical information to competitors can be a great setback.
- Apart from the danger of the insider information leaking out, there is a great danger of the outside elements (such as viruses and worms) entering a corporate network to create havoc.

We can depict this situation as shown in Fig. 9.5.

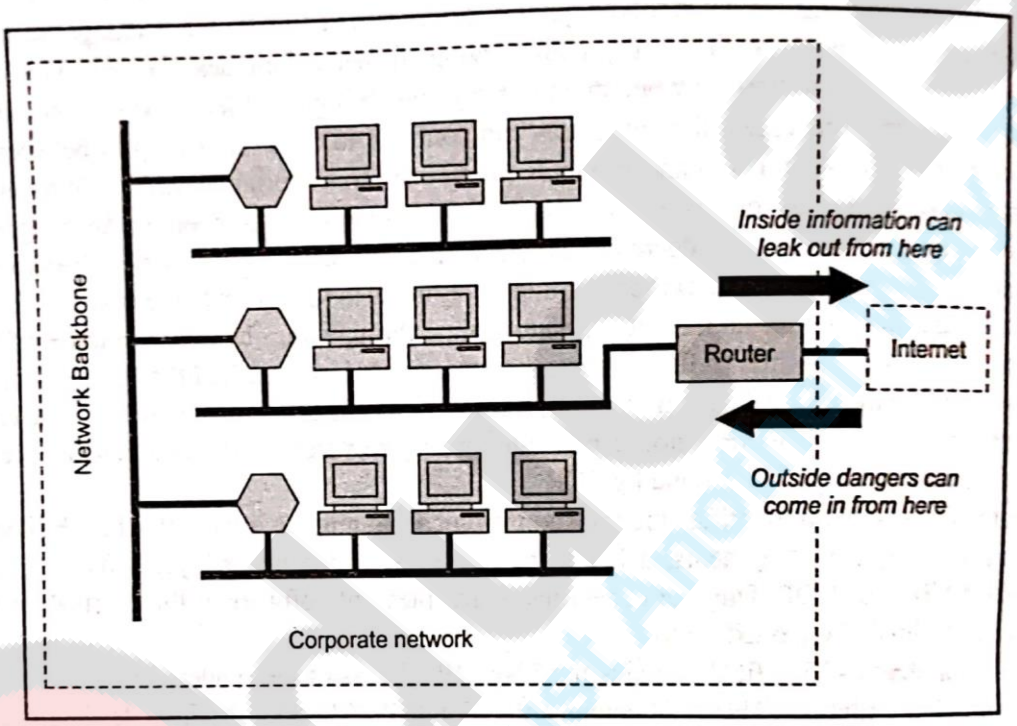

Network Backbone
Inside information can leak out from here
Router
Internet
Outside dangers can come in from here
Corporate network

Fig. 9.5 *Threats from inside and outside a corporate network*

As a result of these dangers, we must have mechanisms which can ensure that the inside information remains inside and also prevent the outsider attackers from entering inside a corporate network. As we know, encryption of information (if implemented properly) renders its transmission to the outside world redundant. That is, even if confidential information flows out of a corporate network, if it is in encrypted form, outsiders cannot make any sense of it. However, encryption does not work in the other direction. Outside attackers can still try to break inside a corporate network. Consequently, better schemes are desired to achieve protection from outside attacks. This is where a **firewall** comes into picture.

Conceptually, a firewall can be compared with a sentry standing outside an important person's house (such as the nation's president). This sentry usually keeps an eye on and physically checks every person that enters into or comes out of the house. If the sentry senses that a person wishing to enter the president's house is carrying a knife, the sentry would not allow the person to enter. Similarly, even if the person does not possess any banned objects, but somehow looks suspicious, the sentry can still prevent that person's entry.

A firewall acts like a sentry. If implemented, it guards a corporate network by standing between the network and the outside world. All traffic between the network and the Internet in either direction must pass through the firewall. The firewall decides if the traffic can be allowed to flow or whether it must be stopped from proceeding further. This is shown in Fig. 9.6.

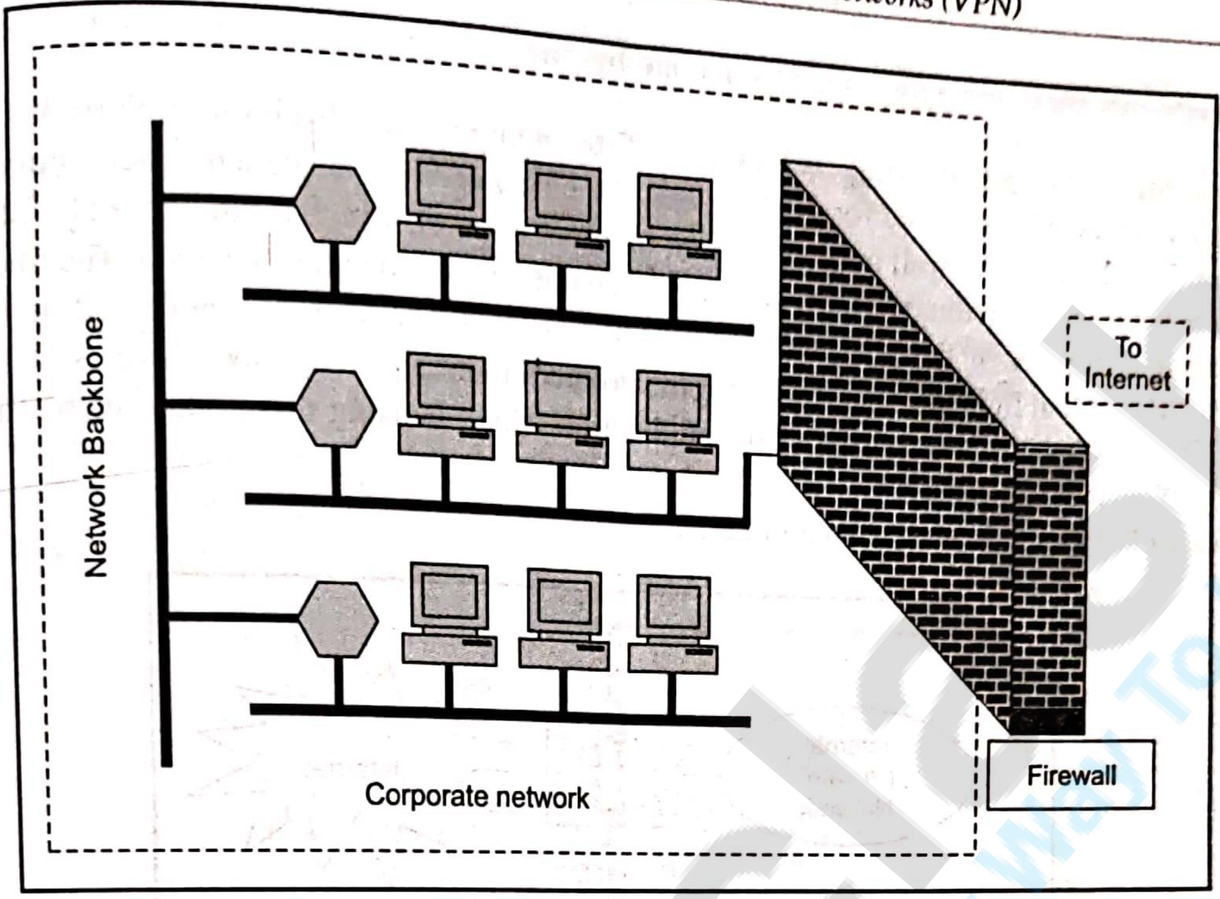


Fig. 9.6 *Firewall*

Of course, technically, a firewall is a specialized version of a router. Apart from the basic routing functions and rules, a router can be configured to perform the firewall functionality, with the help of additional software resources.

The characteristics of a good firewall implementation can be described as follows.

- All traffic from inside to outside and vice versa, must pass through the firewall. To achieve this, all the access to the local network must first be physically blocked and access only via the firewall should be permitted.
- Only the traffic authorized as per the local security policy should be allowed to pass through.
- The firewall itself must be strong enough, so as to render attacks on it useless.

9.3.2 Types of Firewalls

Based on the criteria that they use for filtering traffic, firewalls are generally classified into two types, as shown in Fig. 9.7.

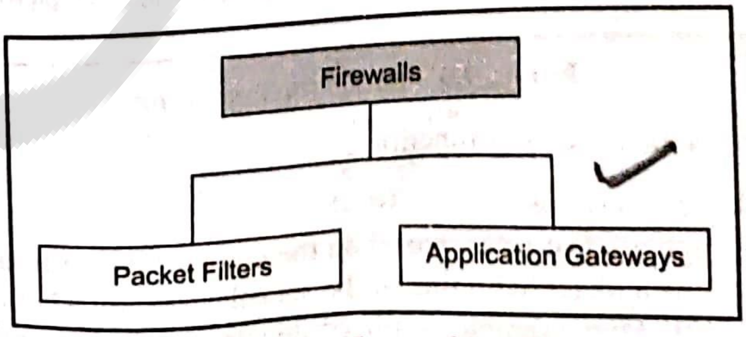


Fig. 9.7 *Types of firewalls*

Let us discuss these two types of firewalls one-by-one.

Packet Filters As the name suggests, a **packet filter** applies a set of rules to each packet and based on the outcome, decides to either forward or discard the packet. It is also called as **screening router** or **screening filter**. Such a firewall implementation involves a router, which is configured to filter packets going in either direction (from the local network to the outside world and vice versa). The filtering rules are based on a number of fields in the IP and TCP/UDP headers, such as source and destination IP addresses, IP protocol field (which identifies if the protocol in the upper transport layer is TCP or UDP), TCP/UDP port numbers (which identify the application which is using this packet, such as email, file transfer or World Wide Web).

The idea of a packet filter is shown in Fig. 9.8.

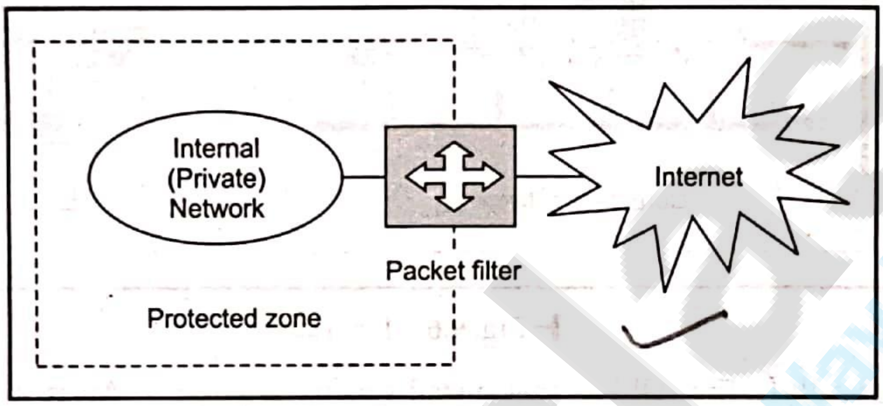


Fig. 9.8 *Packet filter*

Conceptually, a packet filter can be considered as a router that performs three main actions, as shown in Fig. 9.9.

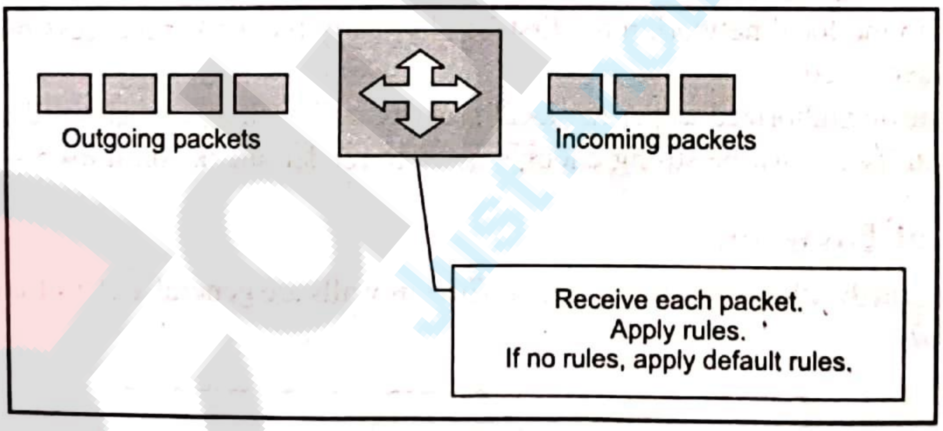


Fig. 9.9 *Packet filter operation*

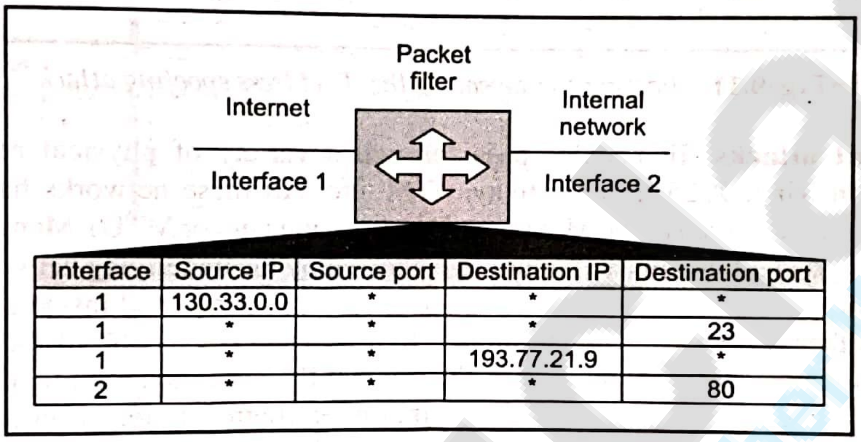
A packet filter performs the following functions.

1. Receive each packet as it arrives.
2. Pass the packet through a set of rules, based on the contents of the IP and transport header fields of the packet. If there is a match with one of the set rules, decide whether to accept or discard the packet based on that rule. For example, a rule could specify: disallow all incoming traffic from an IP address 157.29.19.10 (this IP address is taken just as an example) or disallow all traffic that uses UDP as the higher (transport) layer protocol.

3. If there is no match with any rule, take the default action. The default can be *discard all packets* or *accept all packets*. The former policy is more conservative, whereas the latter is more open. Usually, the implementation of a firewall begins with the default *discard all packets* option and then rules are applied one-by-one to enforce packet filtering.

The chief advantage of the packet filter is its simplicity. The users need not be aware of a packet filter at all. Packet filters are very fast in their operating speed. However, the two disadvantages of a packet filter are the difficulties in setting up the packet filter rules correctly and lack of support for authentication.

Figure 9.10 shows an example where a router can be converted into a packet filter by adding the filtering rules in the form of a table. This table decides which of the packets should be allowed (forwarded) or discarded.



Interface	Source IP	Source port	Destination IP	Destination port
1	130.33.0.0	*	*	*
1	*	*	*	23
1	*	*	193.77.21.9	*
2	*	*	*	80

Fig. 9.10 *Example of packet filter table*

The rules specified in the packet filter work as follows:

1. Incoming packets from network 130.33.0.0 are not allowed. They are blocked as a security precaution.
2. Incoming packets from any external network on the TELNET server port (number 23) are blocked.
3. Incoming packets intended for a specific internal host 193.77.21.9 are blocked.
4. Outgoing packets intended for port 80 (HTTP) are banned. That is, this organization does not want to allow its employees to send requests to the external world (i.e. the Internet) for browsing the Internet.

Attackers can try and break the security of a packet filter by using the following techniques.

1. **IP address spoofing:** An intruder outside the corporate network can attempt to send a packet towards the internal corporate network, with the source IP address set equal to one of the IP addresses of the internal users. This is shown in Fig. 9.11. This attack can be defeated by discarding all the packets that arrive at the incoming side of the firewall, with the source address equal to one of the internal addresses.
2. **Source routing attacks:** An attacker can specify the route that a packet should take as it moves along the Internet. The attacker hopes that by specifying this option, the packet filter can be fooled to bypass its normal checks. Discarding all packets that use this option can thwart such an attack.

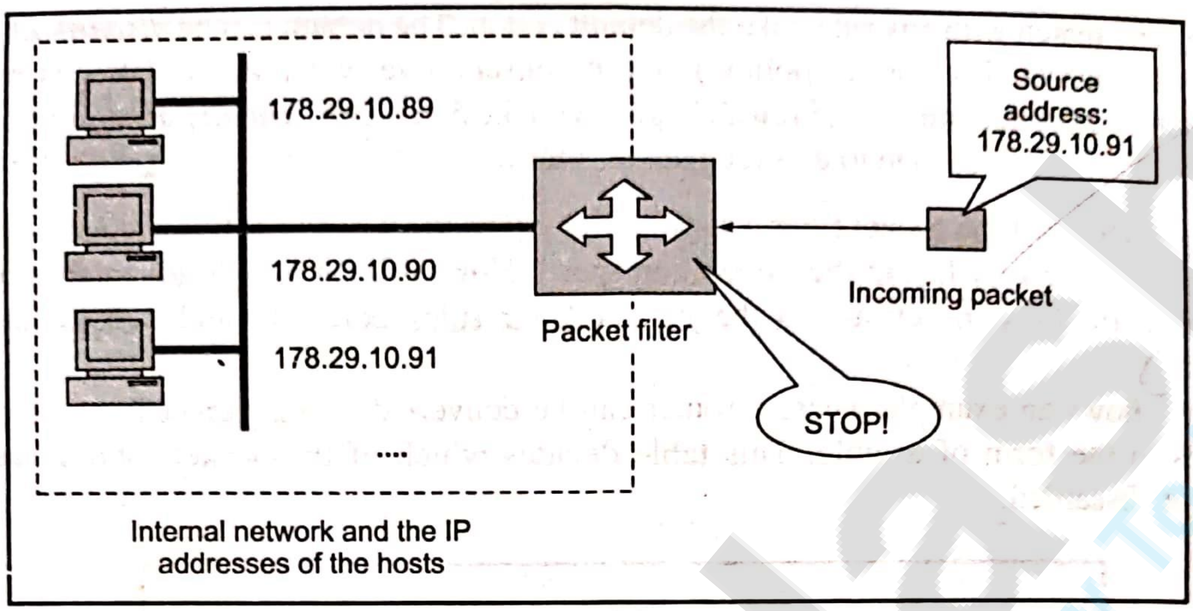

178.29.10.89
178.29.10.90
178.29.10.91
....
Packet filter
Source address: 178.29.10.91
Incoming packet
STOP!
Internal network and the IP addresses of the hosts

Fig. 9.11 *Packet filter defeating the IP address spoofing attack*

3. **Tiny fragment attacks**: IP packets pass through a variety of physical networks, such as Ethernet, Token Ring, X.25, Frame Relay, ATM, etc. All these networks have a pre-defined maximum frame size (called as the Maximum Transmission Unit or MTU). Many times, the size of the IP packet is greater than this maximum size allowed by the underlying network. In such cases, the IP packet needs to be fragmented, so that it can be accommodated inside the physical frame and carried further. An attacker might attempt to use this characteristic of the TCP/IP protocol suite by intentionally creating fragments of the original IP packet and sending them. The attacker feels that the packet filter can be fooled, so that after fragmentation, it checks only the first fragment and does not check the remaining fragments. This attack can be foiled by discarding all the packets where the (upper layer) protocol type is TCP and the packet is fragmented (refer to *identification* and *protocol* fields of an IP packet discussed earlier to understand how we can implement this).

An advanced type of packet filter is called as **dynamic packet filter** or **stateful packet filter**. A dynamic packet filter allows the examination of packets based on the current state of the network. That is, it adapts itself to the current exchange of information, unlike the normal packet filters, which have routing rules hard coded. For instance, we can specify a rule with the help of a dynamic packet filter as follows:

Allow incoming TCP packets only if they are responses to the outgoing TCP packets that have gone through our network.

p the packet.)

Application Gateways An **application gateway** is also called as a **proxy server**. This is because it acts like a proxy (i.e. deputy or substitute) and decides about the flow of application level traffic. The idea is shown in Fig. 9.13.

Application gateways typically work as follows.

1. An internal user contacts the application gateway using a TCP/IP application, such as HTTP or TELNET.
2. The application gateway asks the user about the remote host with which the user wants to set up a connection for actual communication (i.e. its domain name or IP address, etc). The application gateway also asks for the user id and the password required to access the services of the application gateway.
3. The user provides this information to the application gateway.

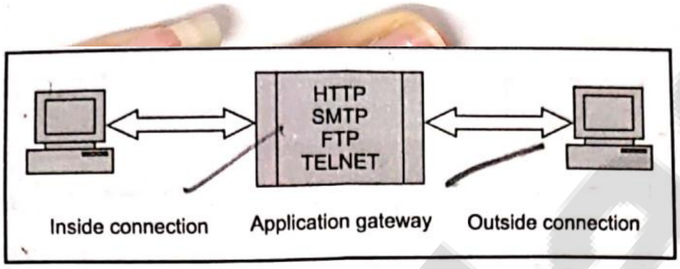


Fig. 9.13 *Application gateway*

4. The application gateway now accesses the remote host on behalf of the user and passes the packets of the user to the remote host. Note that there is a variation of the application gateway, called as **circuit gateway**, which performs some additional functions as compared to those performed by an application gateway. A circuit gateway, in fact, creates a new connection between itself and the remote host. The user is not aware of this and thinks that there is a direct connection between itself and the remote host. Also, the circuit gateway changes the source IP address in the packets from the end user's IP address to its own. This way, the IP addresses of the computers of the internal users are hidden from the outside world. This is shown in Fig. 9.14. Of course, both the connections are shown with a single arrow to stress on the concept, in reality, both the connections are two-ways.

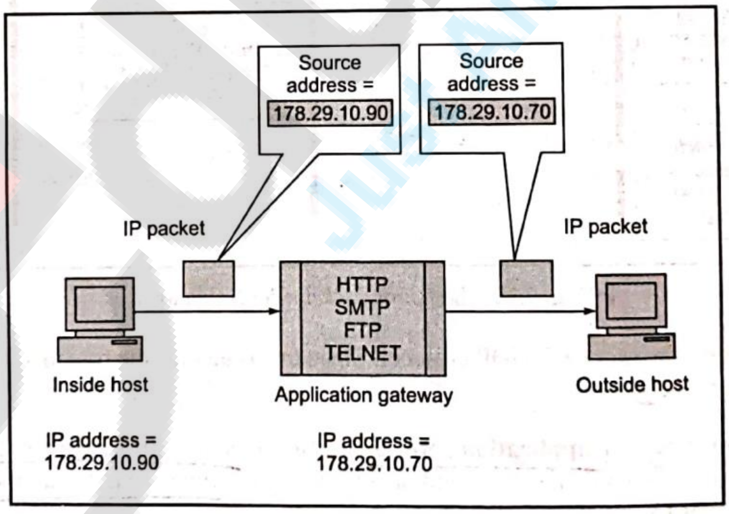


Fig. 9.14 *Circuit gateway operation*

The SOCKS server is an example of the real-life implementation of a circuit gateway. It is a client-server application. The SOCKS client runs on the internal hosts and the SOCKS server runs on the firewall.

5. From here onwards, the application gateway acts like a proxy of the actual end user and delivers packets from the user to the remote host and vice versa.

Application gateways are generally more secure than packet filters, because rather than examining every packet against a number of rules, here we simply detect whether a user is allowed to work with a TCP/IP application or not. The disadvantage of application gateways is the overhead in terms of connections. As we noticed, there are actually two sets of connections now: one between the end user and the application gateway and another between the application gateway and the remote host. The application gateway has to manage these two sets of connections and the traffic going between them. This means that the actual communicating internal host is under an illusion, as illustrated in Fig. 9.15.

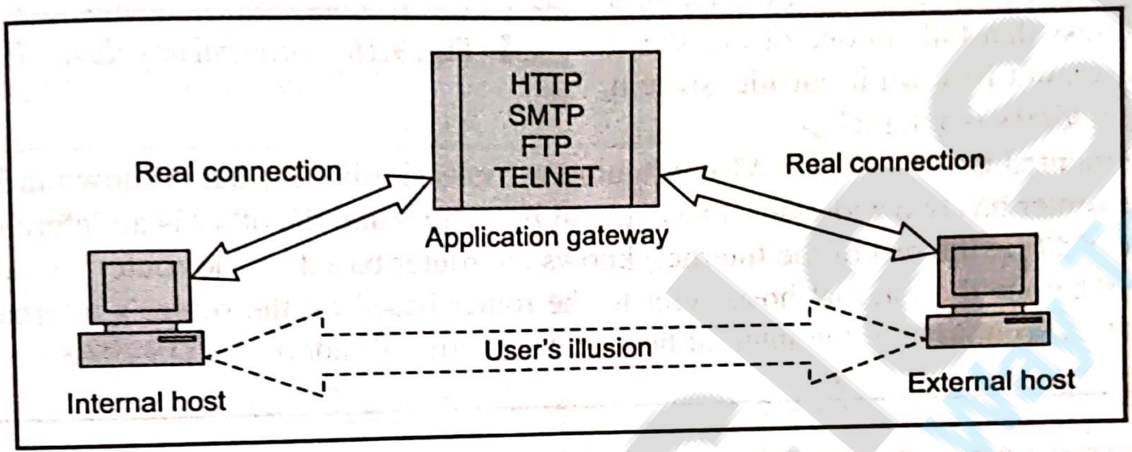


Fig. 9.15 *Application gateway creates an illusion*

An application gateway is also called as **bastion host**. Usually, a bastion host is a very key point in the security of a network. How it functions is explained in more detail in the next section.

Network Address Translation (NAT) One of the interesting jobs done by a firewall or proxy server is to perform **Network Address Translation (NAT)**. The number of people using the Internet from home, office or other places is increasing at a mind boggling rate. Earlier, users would access the Internet via an Internet Service Provider (ISP) for a short time and then disconnect. Thus, the ISP would have a set of IP addresses, from which it would dynamically allocate one IP address to every user for the duration the user was connected to the Internet. Once the user disconnected, the ISP would reallocate that same IP address to another user, who wanted to connect to the Internet now.

However, this situation changed dramatically as the number of people connecting to the Internet increased dramatically. Moreover, people started using the ADSL or cable connections to connect to the Internet, sing the *broadband* technology. Worse yet, people wanted multiple IP addresses for themselves, since they started creating small personal networks. This led to a serious problem of shortage of IP addresses.

9.3.3 Firewall Configurations

In practical implementations, a firewall is usually a combination of packet filters and application (or circuit) gateways. Based on this, there are three possible configurations of firewalls, as shown in Fig. 9.20.

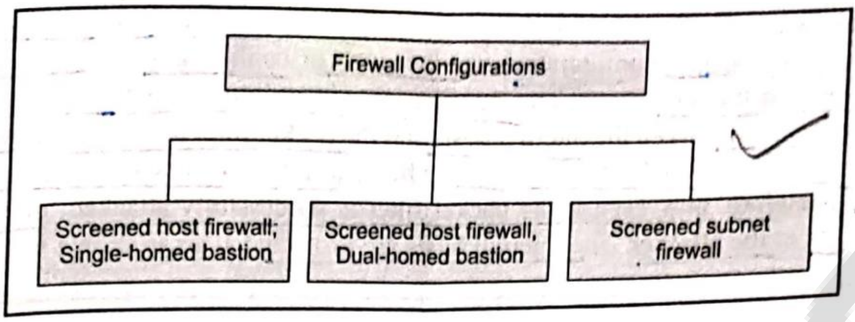

Firewall Configurations
Screened host firewall; Single-homed bastion
Screened host firewall, Dual-homed bastion
Screened subnet firewall

Fig. 9.20 *Firewall configurations*

Let us discuss these possible configurations as follows.

Screened Host Firewall, Single-Homed Bastion In the **Screened host firewall, Single-homed bastion** configuration, a firewall set up consists of two parts: a packet-filtering router and an application gateway. Their purposes are as follows.

- The packet filter ensures that the incoming traffic (i.e. from the Internet to the corporate network) is allowed only if it is destined for the application gateway, by examining the *destination address* field of every incoming IP packet. Similarly, it also ensures that the outgoing traffic (i.e. from the corporate network to the Internet) is allowed only if it is originating from the application gateway, by examining the *source address* field of every outgoing IP packet.
- The application gateway performs authentication and proxy functions, as explained earlier.

This configuration is illustrated in Fig. 9.21.

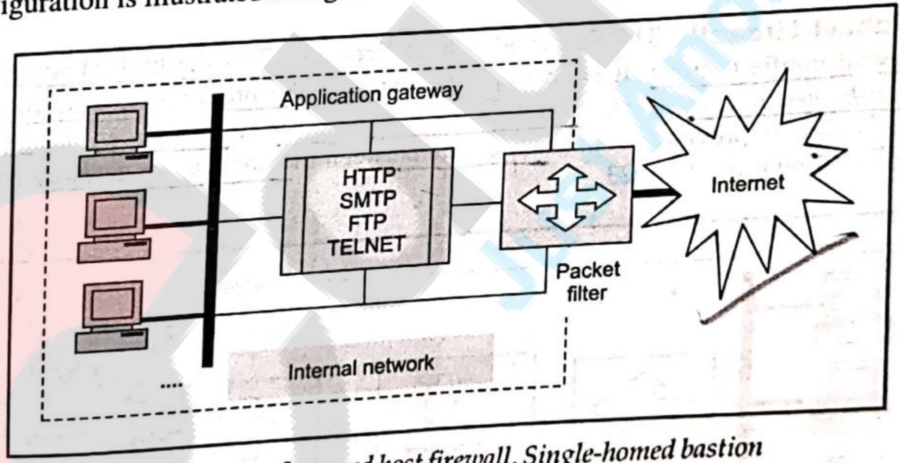


Fig. 9.21 *Screened host firewall, Single-homed bastion*

This configuration increases the security of the network by performing checks at both packet and application levels. This also gives more flexibility to the network administrators to define more granular security policies.

However, as we can see, one big disadvantage here is that the internal users are connected to the application gateway, as well as to the packet filter. Therefore, if the packet filter is somehow successfully attacked and its security compromised, then the whole internal network is exposed to the attacker.

Screened Host Firewall, Dual-Homed Bastion To overcome the drawback of a *screened host firewall, single-homed bastion* configuration, another type of configuration, called as **Screened host firewall, Dual-homed bastion**, exists. This configuration is an improvement over the earlier scheme. Here, direct connections between the internal hosts and the packet filter are avoided. Instead, the packet filter connects only to the application gateway, which, in turn, has a separate connection with the internal hosts. Therefore, now even if the packet filter is successfully attacked, only the application gateway is visible to the attacker. The internal hosts are protected. This is shown in Fig. 9.22.

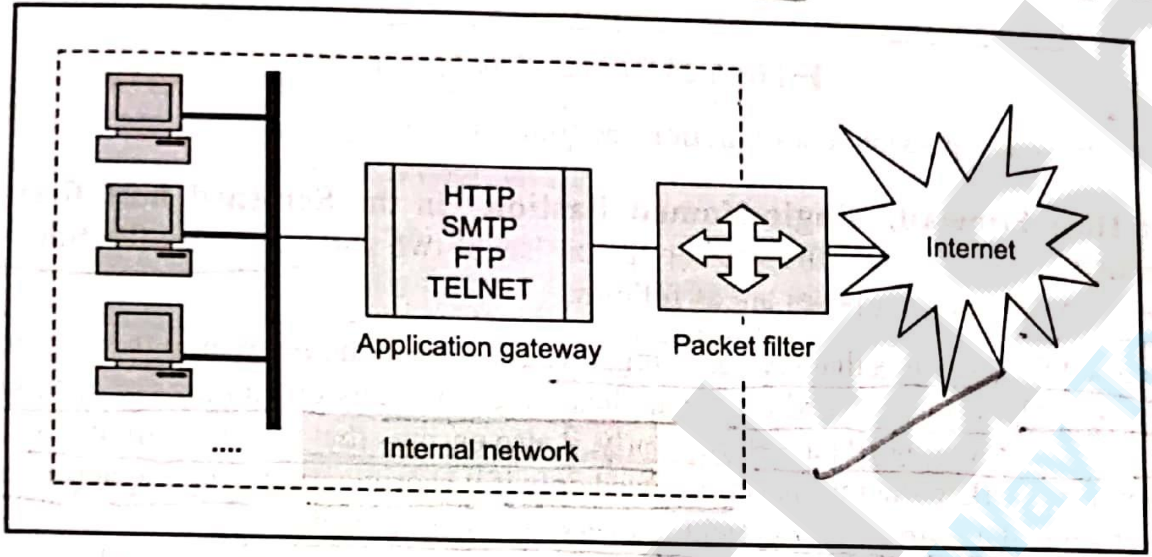


Fig. 9.22 *Screened host firewall, Dual-homed bastion*

Can we think of a scheme, which is even better than this?

Screened Subnet Firewall The **Screened subnet firewall** offers the highest security among the possible firewall configurations. It is an improvement over the previous scheme of *screened host firewall, Dual-homed bastion*. Here, two packet filters are used, one between the Internet and the application gateway, as previously and another one between the application gateway and the internal network. This is shown in Fig. 9.23.

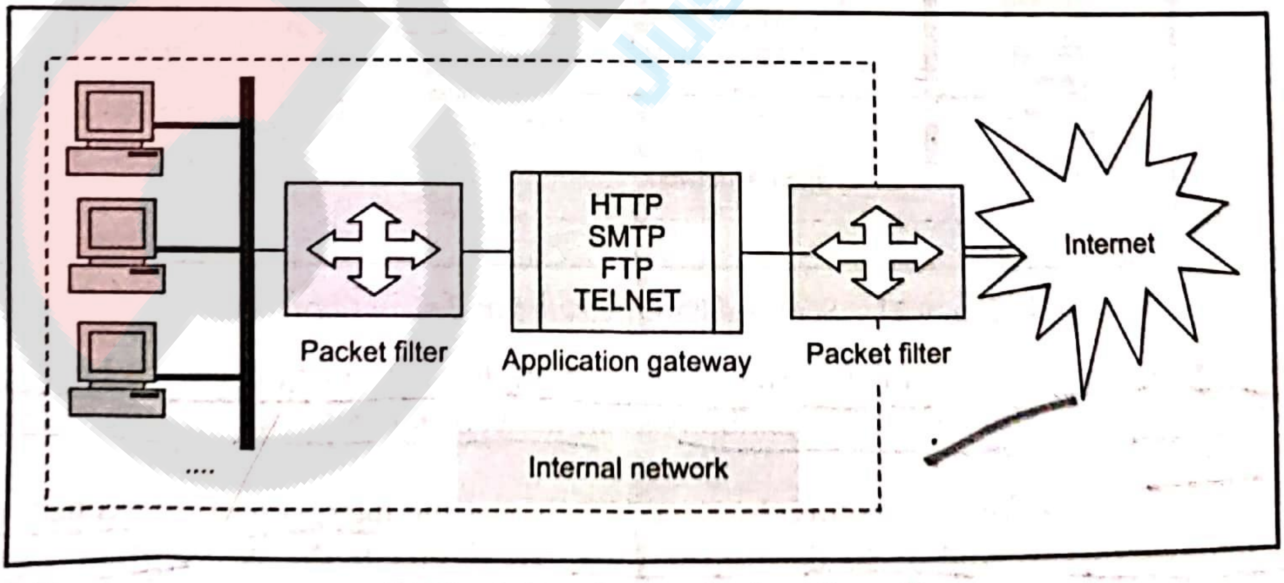
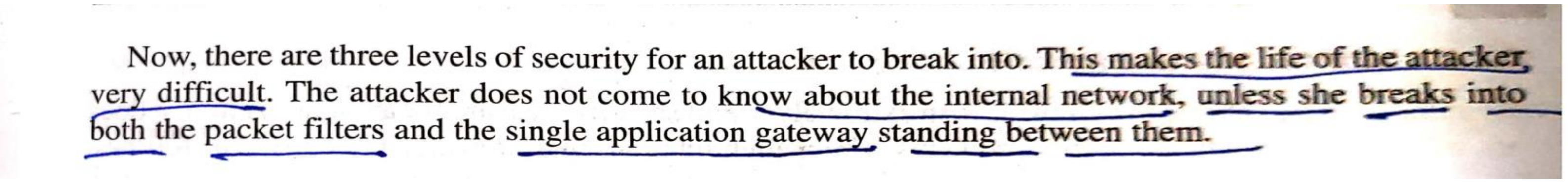


Fig. 9.23 *Screened subnet firewall*



Now, there are three levels of security for an attacker to break into. This makes the life of the attacker very difficult. The attacker does not come to know about the internal network, unless she breaks into both the packet filters and the single application gateway standing between them.

Firewall Architectures and its implementations

1. DUAL-HOMED HOST

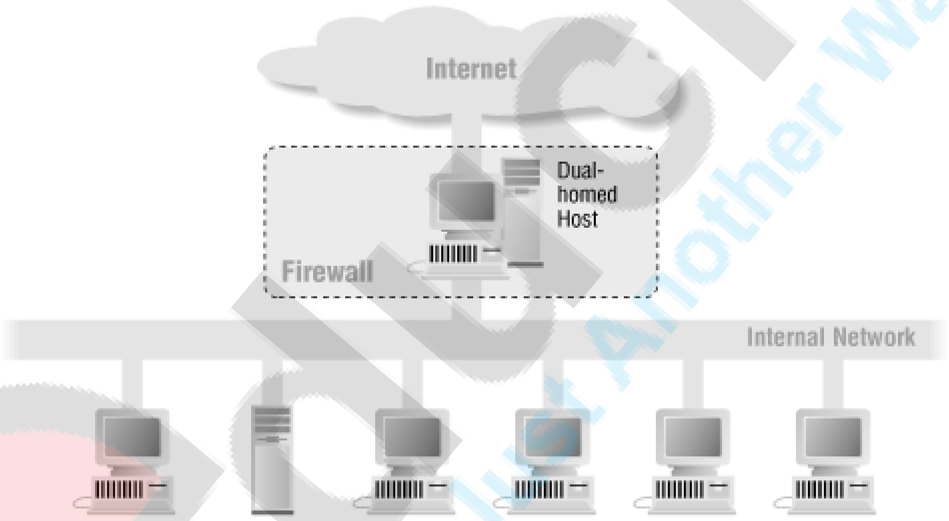
A dual-homed host architecture is built around the dual-homed host computer, a computer that has at least two network interfaces.

Such a host could act as a router between the networks these interfaces are attached to; it is capable of routing IP packets from one network to another.

However, to use a dual-homed host as a firewall, you disable this routing function. Thus, IP packets from one network (e.g., the Internet) are not directly routed to the other network.

Systems inside the firewall can communicate with the dual-homed host, and systems outside the firewall (on the Internet) can communicate with the dual-homed host, but these systems can't communicate directly with each other. IP traffic between them is completely blocked.

The network architecture for a dual-homed host firewall is pretty simple: the dual-homed host sits between, and is connected to, the Internet and the internal network.



Dual-homed hosts can provide a very high level of control. If you aren't allowing packets to go between external and internal networks at all, you can be sure that any packet on the internal network that has an external source is evidence of some kind of security problem.

On the other hand, dual-homed hosts aren't high-performance devices. A dual-homed host has more work to do for each connection than a packet filter does, and correspondingly needs more resources. A dual-homed host won't support as much traffic as an equivalent packet filtering system.

Since a dual-homed host is a single point of failure, it's important to make certain that its host security is absolutely impeccable. An attacker who can compromise the dual-homed host has full access to your site.

A dual-homed host can provide services only by proxying them, or by having users log into the dual-homed host directly.

USES:

A dual-homed host is an appropriate firewall for a situation where:

- Traffic to the Internet is small.
- Traffic to the Internet is not business-critical.
- No services are being provided to Internet-based users.
- The network being protected does not contain extremely valuable data.

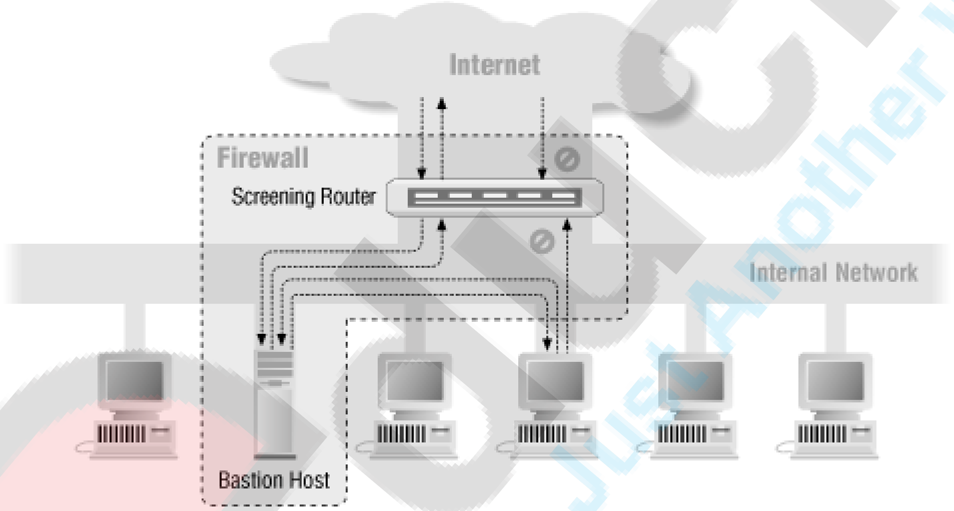
2. SCREENED HOST

A screened host architecture provides services from a host that's attached to only the internal network, using a separate router. In this architecture, the primary security is provided by packet filtering.

The bastion host sits on the internal network. The packet filtering on the screening router is set up in such a way that the bastion host is the only system on the internal network that hosts on the Internet can open connections to.

Even then, only certain types of connections are allowed. Any external system trying to access internal systems or services will have to connect to this host.

The bastion host thus needs to maintain a high level of host security.



Because this architecture allows packets to move from the Internet to the internal networks, it may seem more risky than a dual-homed host architecture, which is designed so that no external packet can reach the internal network.

Also, it's easier to defend a router than it is to defend a host. For most purposes, the screened host architecture provides both better security and better usability than the dual-homed host architecture.

Compared to other architectures, however, such as the screened subnet architecture, there are some disadvantages to the screened host architecture. The major one is that if an attacker manages to break in to the bastion host, nothing is left in the way of network security between the bastion host and the rest of the internal hosts.

USES:

A screened host architecture is appropriate when:

- Few connections are coming from the Internet (in particular, it is not an appropriate architecture if the screened host is a public web server).
- The network being protected has a relatively high level of host security.

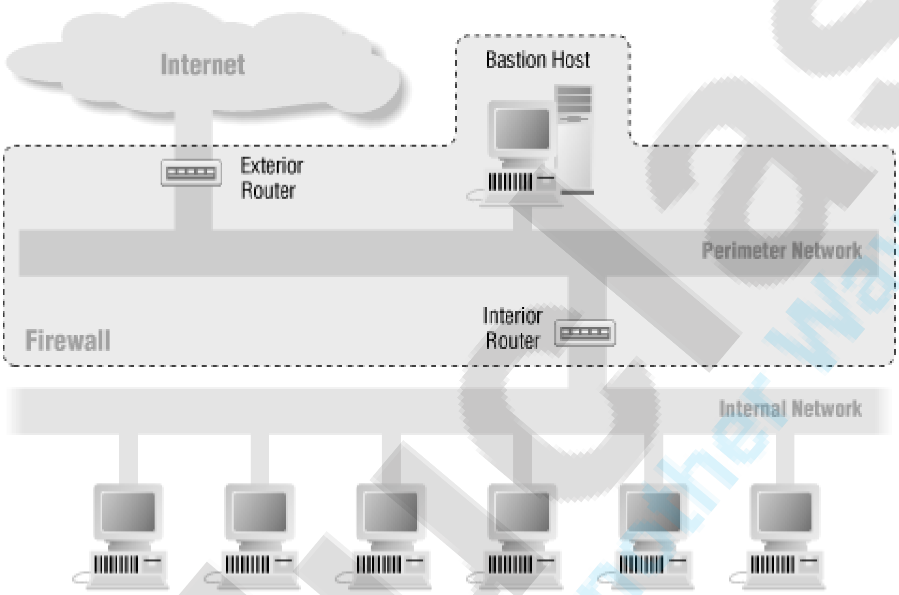
3. SCREENED SUBNET

The screened subnet architecture adds an extra layer of security to the screened host architecture by adding a perimeter network that further isolates the internal network from the Internet.

By their nature, bastion hosts are the most vulnerable machines on your network. Despite your best efforts to protect them, they are the machines most likely to be attacked because they're the machines that can be attacked.

If, as in a screened host architecture, your internal network is wide open to attack from your bastion host, then your bastion host is a very tempting target. No other defenses are between it and your other internal machines.

By isolating the bastion host on a perimeter network, you can reduce the impact of a break-in on the bastion host.



With the simplest type of screened subnet architecture, there are two screening routers, each connected to the perimeter network.

One sits between the perimeter net and the internal network, and the other sits between the perimeter net and the external network (usually the Internet).

To break into the internal network with this type of architecture, an attacker would have to get past both routers.

Even if the attacker somehow broke in to the bastion host, he'd still have to get past the interior router. There is no single vulnerable point that will compromise the internal network.