Firewalls

12

12.1 Definition

The rapid growth of the Internet has led to a corresponding growth of both users and activities in cyberspace. Unfortunately, not all these users and their activities are reputable; thus, the Internet has been increasingly, at least to many individuals and businesses, turning into a "bad Internet." Bad people are plowing the Internet with evil activities that include, among other things, intrusion into company and individual systems looking for company data and individual information that erodes privacy and security. There has, therefore, been a need to protect company systems, and now individual PCs, keeping them out of access from those "bad users" out on the "bad Internet." As companies build private networks and decide to connect them onto the Internet, network security becomes one of the most important concerns network system administrators face. In fact, these network administrators are facing threats from two fronts: the external Internet and the internal users within the company network. So network system administrators must be able to find ways to restrict access to the company network or sections of the network from both the "bad Internet" outside and from unscrupulous inside users.

Such security mechanisms are based on a *firewall*. A firewall is a hardware, a software, or a combination of both that monitors and filters traffic packets that attempt to either enter or leave the protected private network. It is a tool that separates a protected network or part of a network, and now increasingly a user PC, from an unprotected network—the "bad network" like the Internet. In many cases the "bad network" may even be part of the company network. By definition, a "firewall," is a tool that provides a filter of both incoming and outgoing packets. Most firewalls perform two basic security functions:

- Packet filtering based on *accept* or *deny* policy that is itself based on rules of the security policy.
- Application proxy gateways that provide services to the inside users and at the same time protect each individual host from the "bad" outside users.

J.M. Kizza, *Guide to Computer Network Security*, Computer Communications and Networks, DOI 10.1007/978-3-319-55606-2_12

By denying a packet, the firewall actually drops the packet. In modern firewalls, the firewall logs are stored into log files, and the most urgent or dangerous ones are reported to the system administrator. This reporting is slowly becoming real time. We will discuss this shortly.

In its simplest form, a firewall can be implemented by any device or tool that connects a network or an individual PC to the Internet. For example, an Ethernet bridge or a modem that connects to the "bad network" can be set as a firewall. Most firewall products actually offer much more as they actively filter packets from and into the organization network according to certain established criteria based on the company security policy. Most organization firewalls are *bastion host*, although there are variations in the way this is set up. A bastion host is one computer on the organization network with bare essential services, designated and strongly fortified to withstand attacks. This computer is then placed in a location where it acts as a gateway or a choke point for all communication into or out of the organization network to the "bad network." This means that every computer behind the bastion host must access the "bad network" or networks through this bastion host. Figure 12.1 shows the position of a bastion host in an organization network.

For most organizations, a firewall is a network perimeter security, a first line of defense of the organization's network that is expected to police both network traffic inflow and outflow. This perimeter security defense varies with the perimeter of the network. For example, if the organization has an extranet, an extended network consisting of two or more LAN clusters, or the organization has a virtual private network (VPN) (see Sect. 17.4.2), then the perimeter of the organization's network is difficult to define. In this case, then each component of the network should have its own firewall. See Fig. 12.2.

As we pointed out earlier, the accept/deny policy used in firewalls is based on an organization's security policy. The security policies most commonly used by organizations vary ranging from completely disallowing some traffic to allowing some of the traffic or all the traffic. These policies are consolidated into two commonly used firewall security policies [1]:

- Deny-everything-not-specifically-allowed which sets the firewall in such a way that it denies all traffic and services except a few that are added as the organization needs develop.
- Allow-everything-not-specifically-denied which lets in all the traffic and services except those on the "forbidden" list which is developed as the organization's dislikes grow.

Based on these policies, the following design goals are derived:

- All traffic into and out of the protected network must pass through the firewall.
- Only authorized traffic, as defined by the organizational security policy, in and out of the protected network, will be allowed to pass.
- The firewall must be immune to penetration by use of a trusted system with secure operating system.

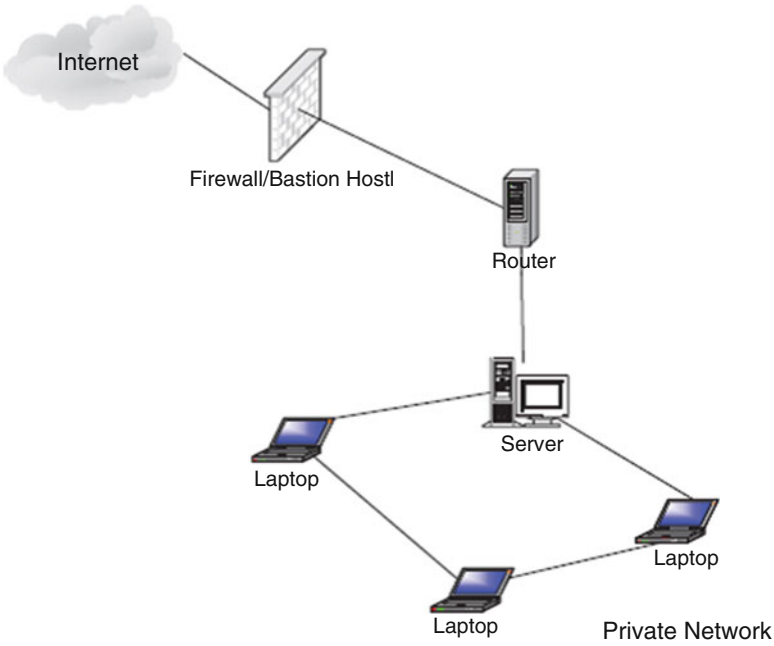

Fig. 12.1 Bastion host between a private network and the "bad network"

When these policies and goals are implemented in a firewall, then the firewall is supposed to [1]:

- Prevent intruders from entering and interfering with the operations of the organization's network. This is done through restricting which packets can enter the network based on IP addresses or port numbers.
- Prevent intruders from deleting or modifying information either stored or in motion within the organization's network.
- Prevent intruders from acquiring proprietary organization information.
- Prevent insiders from misusing the organization resources by restricting unauthorized access to system resources.
- Provide authentication, although care must be taken because additional services to the firewall may make it less efficient.
- Provide endpoints to the VPN.

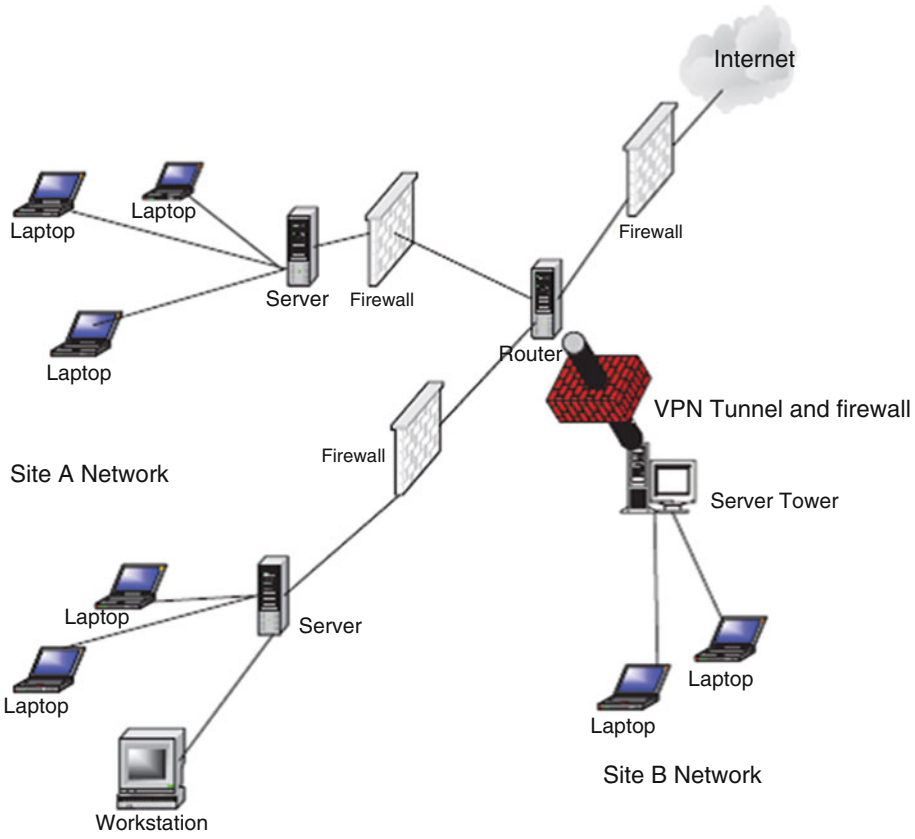

Fig. 12.2 Firewalls in a changing parameter security

12.2 Types of Firewalls

Firewalls are used very widely to offer network security services. This has resulted in a large repertoire of firewalls. To understand the many different types of firewalls, we need only to look at the kind of security services firewalls offer at different layers of the TCP/IP stack.

As Table 12.1 shows, firewalls can be set up to offer security services to many TCP/IP layers. The many types of firewalls are classified based on the network layer it offers services in and the types of services offered.

The first type is the *packet inspection or filtering router*. This type of firewall uses a set of rules to determine whether to forward or block individual packets. A packet inspection router could be a simple machine with multiple network interfaces or a sophisticated one with multiple functionalities. The second type is the *application inspection or proxy server*. The proxy server is based on specific

Table 12.1 Firewall services based on network protocol layers

Layer	Firewall services
Application	Application-level gateways, encryption, SOCKS proxy server
Transport	Packet filtering (TCP, UDP, ICMP)
Network	NAT, IP filtering
Data link	MAC address filtering
Physical	May not be available

application daemons to provide authentication and to forward packets. The third type is the *authentication and virtual private networks* (VPN). A VPN is an encrypted link in a private network running on a public network. The fourth firewall type is the *small office or home* (SOHO) firewall, and the fifth is the network address translation (NAT).

12.2.1 Packet Inspection Firewalls

Packet filter firewalls, the first type of firewalls, are routers that inspect the contents of the source or destination addresses and ports of incoming or outgoing TCP, UDP, and ICMP packets being sent between networks and accept or reject the packet based on the specific packet policies set in the organization's security policy. Recall that a router is a machine that forwards packets between two or more networks. A packet inspection router, therefore, working at the network level, is programmed to compare each packet to a list of rules set from the organization's security policy, before deciding if it should be forwarded or not. Data is allowed to leave the system only if the firewall rules allow it.

To decide whether a packet should be passed on, delayed for further inspection, or dropped, the firewall looks through its set of rules for a rule that matches the contents of the packet's headers. If the rule matches, then the action to deny or allow is taken; otherwise, an alternate action of sending an ICMP message back to the originator is taken.

Two types of packet filtering are used during packet inspection: *static or stateless filtering* in which a packet is filtered in isolation of the context it is in, and *stateful filtering* in which a packet is filtered actually based on the context the packet is in. The trend now for most inspection firewalls is to use stateful filtering.

The *static or stateless filtering* is a full-duplex communication bastion server allowing two-way communication based on strict filtering rules. Each datagram entering the server either from the "bad" network outside the company network or from within the network is examined based on the preset filtering rules. The rules apply only to the information contained in the packet, and anything else like the state of the connection between the client and the server is ignored.

The *stateful filtering* is also a full-duplex communication bastion server. However, unlike the straight packet filtering firewall, this filters every datagram entering the server both from within and outside the network based on the context which

requires a more complex set of criteria and restrictions. For each packet, the firewall examines the date and state of connection between the client and the server. Because this type of filtering pays attention to the data payload of each packet, it is, therefore, more useful and of course more complex. Examination of the data part of the packet makes it useful in detecting questionable data such as attachments and data from hosts not directly connected to the server. Requests from or to third-party hosts and server to server are strictly inspected against the rule base and logged by the firewall.

Whether static or stateful, the rules a filtering server follows are defined based on the organization's network security policy, and they are based on the following information in the packet [2, 3]:

- Source address: All outgoing packets must have a source address internal to the network. Inbound packets must never have source addresses that are internal.
- Destination address: Similarly, all outgoing packets must not have a destination address internal to the network. Any inbound packet must have a destination address that is internal to the network.
- TCP or UDP source and destination port number.
- ICMP message type.
- Payload data type.
- Connection initialization and datagram using TCP ACK bit.

As Table 12.1 shows, packet inspection based on IP addresses, port numbers, ACK, and sequence numbers, on TCP, UDP, and ICMP headers and on applications, may occur at any one of the following TCP/IP and ISO stack layers:

- The *link layer* provides physical addressing of devices on the same network. Firewalls operating on the link layer usually drop packets based on the *media access control* (MAC) addresses of communicating hosts.
- The *network layer* contains the *Internet protocol* (IP) headers that support addressing across networks. IP headers are inspected.
- The *transport layer* contains TCP, UDP, and ICMP headers and provides data flows between hosts. Most firewalls operate at the network and transport layer and inspect these headers.
- The *application layer* contains application-specific protocols like HTTP, FTP, and SET. Inspection of application-specific protocols can be computationally expensive because more data needs to be inspected.

Let us now look at the different ways of implementing the filtering firewall based on IP address, TCP/UDP port numbers, sequence numbers, and ACK filtering.

12.2.1.1 IP Address Filtering

IP address filtering rules are used to control traffic into and out of the network through the filtering of both source and destination IP addresses. Since in a stateless filter, no record is kept, the filter does not remember any packet that has passed

Table 12.2 Destination IP filtering

Application protocol	Source IP	Destination IP	Action
HTTP	Any	198.124.1.0	Allow
Telnet	Any	198.213.1.1	Deny
FTP	Any	198.142.0.2	Allow

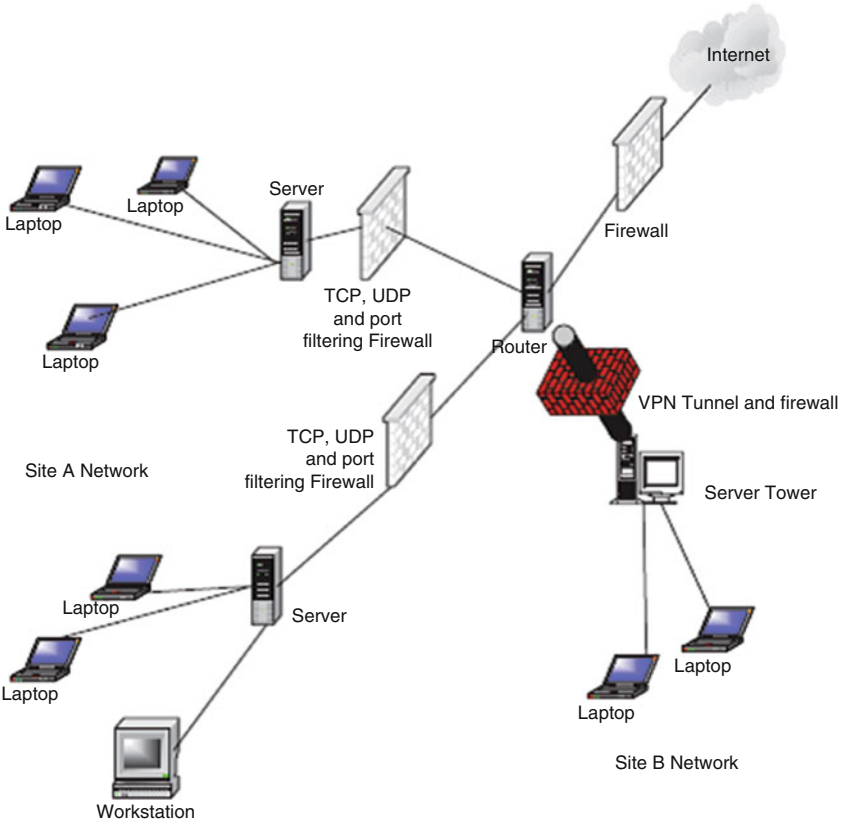

Fig. 12.3 TCP, UDP, and port number filtering firewall

through it. This is a weakness that can be exploited by hackers to do IP spoofing. Table 12.2 shows rules that filter based on IP destination, and Fig. 12.3 shows a TCP, UDP, and port number filtering firewall.

12.2.1.2 TCP and UDP Port Filtering

Although IP address header filtering works very well, it may not give the system administrator enough flexibility to allow users from a trusted network to access specific services from a server located in the "bad network" and vice versa. For example, we may not want users from the "bad network" to telnet into any trusted network host, but the administrator may want to let them access the Web services

Table 12.3 Filtering rules based on TCP and UDP destination port numbers

Application	Protocol	Destination port number	Action
HTTP	TCP	80	Allow
SSL	UDP	443	Deny
Telnet	TCP	23	Allow

that are on the same or another machine. To leave a selective but restricted access to that machine, the administrator has to be able to set filters according to the TCP or UDP port numbers in conjunction with the IP address filters. Table 12.3 illustrates the filtering rules based on TCP and UDP ports number filtering.

Unfortunately, as Eric Hall [4] points out, there are a several problems with this approach. First, it is not easy to know what port numbers the servers that you are trying to access are running on. As Hall observes, modern-day servers such as HTTP and Gopher are completely configurable in this manner, allowing the user to run them on any port of choice. If this type of filtering is implemented, then the network users will not be able to access those sites that do not use the "standard" port numbers prescribed. In addition to not being able to pinpoint to a "standard" port number, there is also a potential of some of the incoming response packets coming from an intruder port 80.

12.2.1.3 Packet Filtering Based on Initial Sequence Numbers (ISNs) and Acknowledgment (ACK) Bits

A fundamental notion in the design and reliability of the TCP protocol is a sequence number. Every TCP connection begins with a three-way handshaking sequence that establishes specific parameters of the connection. The connection parameters include informing the other host of the sequence numbers to be used. The client initiates the three-way handshake connection request by not only setting the synchronization (SYN) flag but also indicating the *initial sequence number* (ISN) that it will start within addressing data bytes, the octets. This ISN is placed in the sequence number field.

Upon receipt of each octet, the server responds by setting the header flags SYN and ACK; it also sets its ISN in the sequence number field of the response, and it updates the sequence number of the next octet of data it expects from the client.

The acknowledgment is cumulative so that an acknowledgment of sequence number n indicates that all octets up to but not including n have been received. This mechanism is good for duplicate detection in the presence of retransmission that may be caused by replays. Generally, the numbering of octets within a packet is that the first data octet immediately following the header is the lowest numbered, and the following octets are numbered consecutively. For the connection to be maintained, every subsequent TCP packet in an exchange must have its octets' ACK bits set for the connection to be maintained. So the ACK bit indicates whether a packet is requesting a connection or a connection has been made. Packets with 0 in the ACK field are requesting for connections, while those with a 1 have ongoing connections. A firewall can be configured to allow packets with ACK bit 1 to access only specified ports and only in designated directions since hackers can

Table 12.4 Rules for filtering based on ACK field bit

Sequence number	IP Destination address	Port number	ACK	Action
15	198.123.0.1	80	0	Deny
16	198.024.1.1	80	1	Allow

insert a false ACK bit of 1 into a packet. This makes the host think that a connection is ongoing. Table 12.4 shows the rules to set the ACK field.

Access control can be implemented by monitoring these ACK bits. Using these ACK bits, one can limit the types of incoming data to only response packets. This means that a remote system or a hacker cannot initiate a TCP connection at all, but can only respond to packets that have been sent to it.

However, as Hall notes, this mechanism is not hacker proof since monitoring TCP packets for the ACK bit doesn't help at all with UDP packets, as they don't have any ACK bit. Also there are some TCP connections such as FTP that initiate connections. Such applications then cannot work across a firewall based on ACK bits.

12.2.1.4 Problems with Packet Filtering Firewalls

Although packet filtering, especially when it includes a combination of other preferences, can be effective, it, however, suffers from a variety of problems including the following:

- UDP port filtering: UDP was designed for unreliable transmissions that do not require or benefit from negotiated connections such as broadcasts, routing protocols, and advertise services. Because it is unreliable, it does not have an ACK bit; therefore, an administrator cannot filter it based on that. Also an administrator cannot control where the UDP packet was originated. One solution for UDP filtering is to deny all incoming UDP connections but allow all outgoing UDP packets. Of course, this policy may cause problems to some network users because there are some services that use UDP such as NFS, NTP, DNS, WINS, NetBIOS over TCP/IP, and NetWare/IP and client applications such as Archie and IRC. Such a solution may limit access to these services for those network users.
- Packet filter routers don't normally control other vulnerabilities such as SYN flood and other types of host flooding.
- Packet filtering does not control traffic on VPN.
- Filtering, especially on old firewalls, does not hide IP addresses of hosts on the network inside the filter but lets them go through as outgoing packets where an intruder can get them and target the hosts.
- They do not do any checking on the legitimacy of the protocols inside the packet.

12.2.2 Application Proxy Server: Filtering Based on Known Services

Instead of setting filtering based on IP addresses, port numbers, and sequence numbers, which may block some services from users within the protected network trying to access specific services, it is possible to filter traffic based on popular services in the organization. Define the filters so that only packets from well-known and popularly used services are allowed into the organization network, and reject any packets that are not from specific applications. Such firewall servers are known as *proxy servers*.

A proxy server, sometimes just an application firewall, is a machine server that sits between a client application and the server offering the services the client application may want. It behaves as a server to the client and as a client to the server, hence a proxy, providing a higher level of filtering than the packet filter server by examining individual application packet data streams. As each incoming data stream is examined, an appropriate application proxy, a program, similar to normal system daemons, is generated by the server for that particular application. The proxy inspects the data stream and makes a decision of either to forward, drop, or refer for further inspection. Each one of these special servers is called a *proxy server*. Because each application proxy is able to filter traffic based on an application, it is able to log and control all incoming and outgoing traffic and therefore offer a higher level of security and flexibility in accepting additional security functions like user-level authentication, end-to-end encryption, intelligent logging, information hiding, and access restriction based on service types [1].

A proxy firewall works by first intercepting a request from a host on the internal network and then passing it on to its destination, usually the Internet. But before passing it on, the proxy replaces the IP source address in the packet with its own IP address and then passes it on. On receipt of packet from an external network, the proxy inspects the packet, replaces its own IP destination address in the packet with that of the internal host, and passes it on to the internal host. The internal host does not suspect that the packet is from a proxy. Figure 12.4 shows a dual-homed proxy server. Modern proxy firewalls provide three basic operations [5]:

- Host IP address hiding: When the host inside the trusted network sends an application request to the firewall and the firewall allows the request through to the outside Internet, a sniffer just outside the firewall may sniff the packet, and it will reveal the source IP address. The host then may be a potential victim for attack. In IP address hiding, the firewall adds to the host packet its own IP header. So that the sniffer will only see the firewall's IP address. So application firewalls then hide source IP addresses of hosts in the trusted network.
- Header destruction: An automatic protection that some application firewalls use to destroy outgoing packet TCP, UDP, and IP headers and replace them with its own headers so that a sniffer outside the firewall will see only the firewall's IP address. In fact, this action stops all types of TCP, UDP, and IP header attacks.
- Protocol enforcement: Since it is common in packet inspection firewalls to allow packets through based on common port numbers, hackers have exploited this by

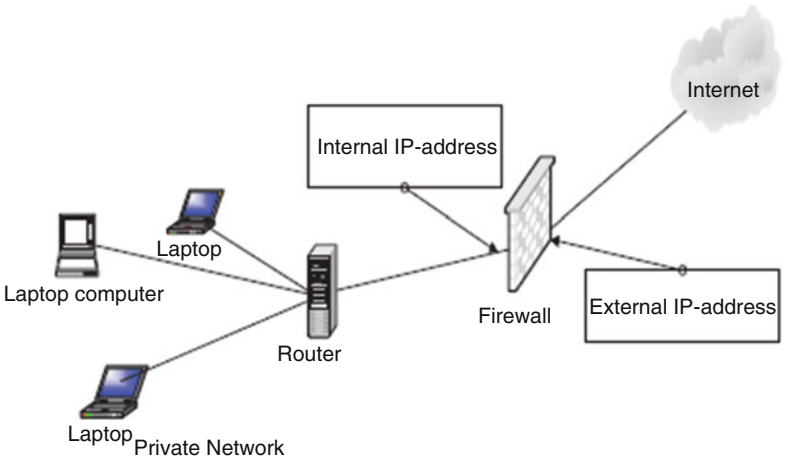

Fig. 12.4 A dual-homed proxy server

port spoofing where they hackers penetrate a protected network host using common used and easily allowed port numbers. With an application proxy firewall, this is not easy to do because each proxy acts as a server to each host, and since it deals with only one application, it is able to stop any port spoofing activities.

An example of a proxy server is a Web application firewall server. Popular Web applications are filtered based on their port numbers as below:

- HTTP (port 80)
- FTP (port 20 and 21)
- SSL (port 443)
- Gopher (port 70)
- Telnet (port 23)
- Mail (port 25)

For newer application firewall, the following proxies are also included: HTTP/Secure HTTP, FTP, SSL, Gopher, email, telnet, and others. This works for both incoming and outgoing requests.

Proxy firewalls fall into two types: application and SOCKS proxies [7, 8].

12.2.2.1 Application Proxy

Application-level proxies automate the filtering and forwarding processes for the client. The client application initiates the process by contacting the firewall. The daemon proxy on the firewall picks up the request, processes it, and if it is acceptable, connects it to the server in the "bad network" (the outside world). If there is any response, it then waits and returns the data to the client application.

As we pointed out earlier, application level proxies offer a higher level of security because in handling all the communications, they can log every detail of the process, including all URLs visited and files downloaded. They can also be used as virus scans, where possible, and language filters for inappropriate content. At login, they can authenticate applications as well as users through a detailed authentication mechanism that includes a one-time password. Also since users do not have direct access to the server, it makes it harder for the intruder to install backdoors around the security system.

Traditional filter firewalls work at a network level to address network access control and block unauthorized network-level requests and access into the network. Because of the popularity of application level services such as e-mail and Web access, application proxy firewalls have become very popular to address application layer security by enforcing requests within application sessions. For example, a Web application firewall specifically protects the Web application communication Web protocol.

There are two models followed in designing an application firewall: a positive security model, which enforces positive behavior; and a negative security model, which blocks recognized attacks [6].

Positive Security Model

A positive security model enforces positive behavior by learning the application logic and then building a security policy of valid known requests as a user interacts with the application. The approach has the following steps [9]:

- The initial policy contains a list of valid starting conditions which the user's initial request must match before the user's session policy is created.
- The application firewall examines the requested services in detail. For example, if it is a Web page download, the page links and drop-down menus and form fields are examined before a policy of all allowable requests that can be made during the user's session is built.
- User requests are verified as valid before being passed to the server. Requests not recognized by the policy are blocked as invalid requests.
- The session policy is destroyed when the user session terminates. A new policy is created for each new session.

Negative Security Model

Unlike the positive model which creates a policy based on user behavior, a negative security model is based on a predefined database of "unacceptable" signatures. The approach again is as follows:

- Create a database of known attack signatures.
- Recognized attacks are blocked, and unknown requests (good or bad) are assumed to be valid and passed to the server for processing.
- All users share the same static policy.

Application firewalls work in real time to address security threats before they reach either the application server or the private network.

12.2.2.2 SOCKS Proxy

A SOCKS proxy is a circuit-level daemon server that has limited capabilities in a sense that it can only allow network packets that originate from non-prohibited sources without looking at the content of the packet itself. It does this by working like a switchboard operator who cross-wires connections through the system to another outside connection without minding the content of the connection, but pays attention only to the legality of the connection. Another way to describe SOCKS servers is to say that these are firewall servers that deal with applications that have protocol behaviors that cannot be filtered. Although they let through virtually all packets, they still provide core protection for application firewalls such as IP hiding and header destruction.

They are faster than application-level proxies because they do not open up the packets, and although they cannot provide for user authentication, they can record and trace the activities of each user as to where he or she is connected to. Figure 12.5 shows a proxy server.

12.2.3 Virtual Private Network (VPN) Firewalls

A VPN, as we will see in Chap. 17, is a cryptographic system including Point-to-Point Tunneling Protocol (PPTP), Layer 2 Tunneling Protocol (L2TP), and IPSec that carry Point-to-Point Protocol (PPP) frames across an Internet with multiple data links with added security. VPNs can be created using a single remote computer connecting on to a trusted network or connecting two corporate network sites. In either case and at both ends of the tunnels, a VPN server can also act as a firewall server. Most firewall servers, however, provide VPN protection which runs in parallel with other authentication and inspection regimes on the server. Each packet arriving at a firewall is then passed through an inspection and authentication module or a VPN module. See Fig. 12.6.

The advantages of a VPN over non-VPN connections like standard Internet connections are as follows:

- VPN technology encrypts its connections.
- Connections are limited to only machines with specified IP addresses.

12.2.4 Small Office or Home (SOHO) Firewalls

A SOHO firewall is a relatively small firewall that connects a few personal computers via a hub, a switch, a bridge, and even a router on one side and connecting to a broadband modem like DSL or cable on the other. See Fig. 12.7. The configuration can be in a small office or a home.

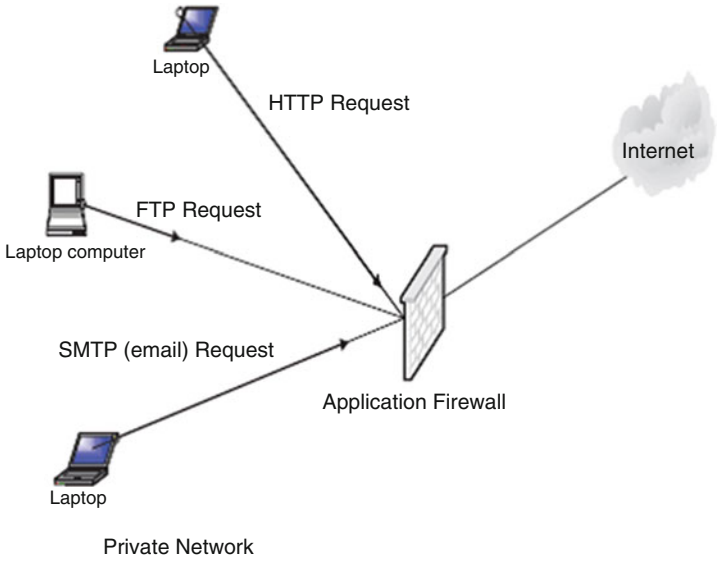

Fig. 12.5 A proxy firewall server

In a functioning network, every host is assigned an IP address. In a fixed network where these addresses are static, it is easy for a hacker to get hold of a host and use it to stage attacks on other hosts within and outside the network. To prevent this from happening, a NAT filter can be used. It hides all inside host TCP/IP information. A NAT firewall actually functions as a proxy server by hiding identities of all internal hosts and making requests on behalf of all internal hosts on the network. This means that to an outside host, all the internal hosts have one public IP address that of the NAT.

When the NAT receives a request from an internal host, it replaces the host's IP address with its own IP address. Inward bound packets all have the NAT's IP address as their destination address. Figure 12.8 shows the position of a NAT firewall.

12.3 Configuration and Implementation of a Firewall

There are actually two approaches to configuring a firewall to suit the needs of an organization. One approach is to start from nothing and make the necessary information gathering to establish the needs and requirements of the organization. This is a time-consuming approach and probably more expensive. The other approach is what many organizations do and takes a shortcut and installs a vendor firewall already loaded with features. The administrator then chooses the features that best meet the established needs and requirements of the organization.

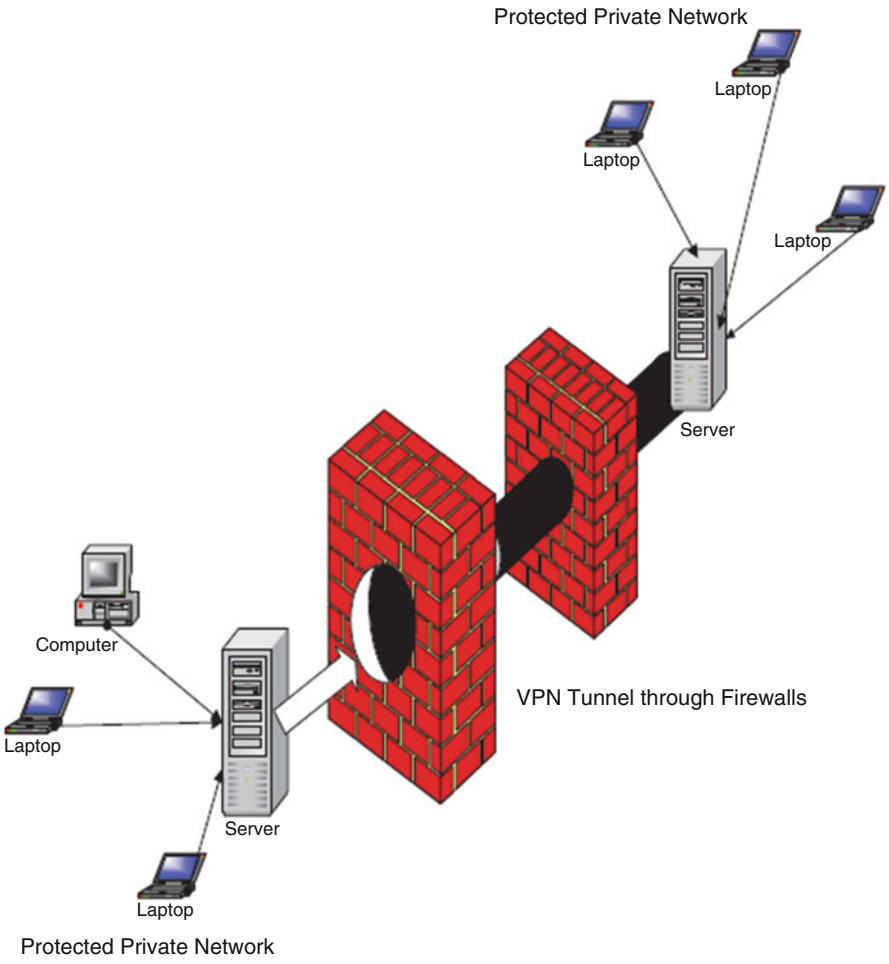

Fig. 12.6 VPN connections and firewalls

Whether the organization is doing an in-house design of its own firewall or buying it off the shelf, the following issues must be addressed first [7]:

- Technical capacity: Whether large or small, organizations embarking on installation of firewalls need some form of technical capacity. Such capacity may be outsourced if it suits the organization.
- Security review: Before an organization can install a firewall, there must be security mechanisms based on a security policy to produce a prioritized list of security objectives.
- Auditing requirements: Based on the security policy, auditing frequency, and what must be in the audit, for example, the degree of logging needed and the

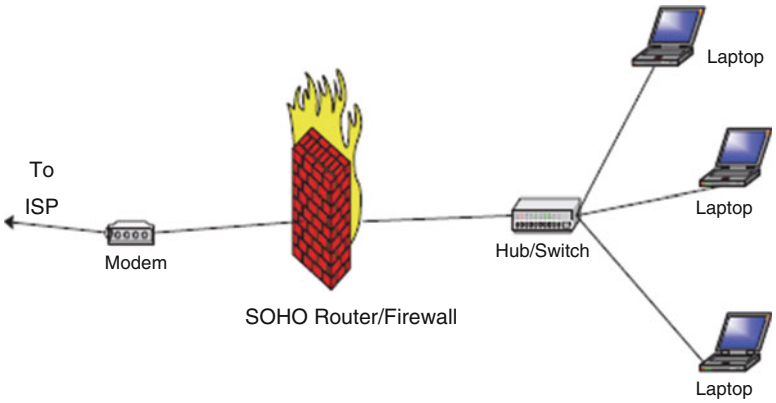

Fig. 12.7 A SOHO firewall

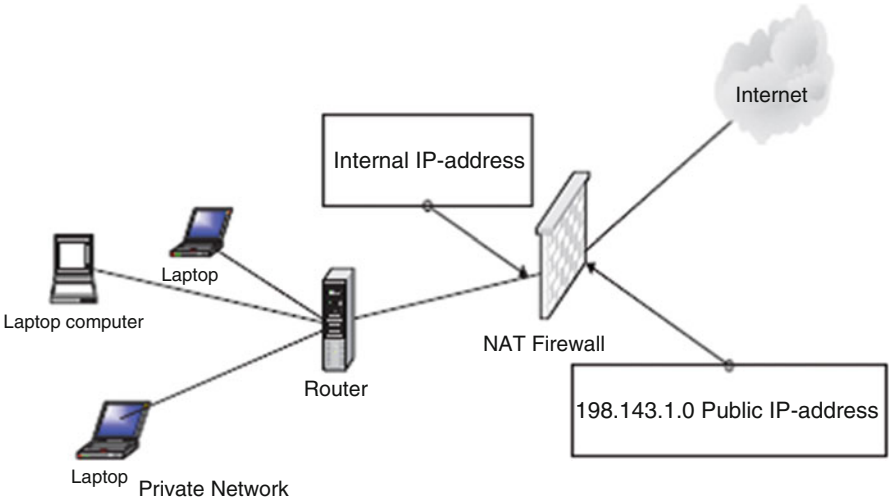

Fig. 12.8 A NAT firewall

details that are cost-effective and thorough. The details included guidelines for recordings, especially if the organization has plans of pursuing security incidents in courts of law.

- Filtering and performance requirements: Decide on the acceptable trade-off between security and performance for the organization. Then use this trade-off to set the level of filtering that meets that balance.
- Authentication: If authentication for outbound sessions is required, then install it and make sure that users are able to change their passwords.

- Remote access: If accepting remote access is to be allowed, include the requirements for authentication and encryption of those sessions. Also consider using VPN to encrypt the session. Many firewalls come with a VPN rolled in.
- Application and network requirements: Decide on the type of network traffic to be supported; whether network address translation (NAT), static routing, or dynamic routing are needed; and whether masquerading a block of internal addresses is sufficient instead of NAT. As Fennelly [10] puts it, a poor understanding of the requirements can lead to implementing a complicated architecture that might not be necessary.
- Decide on the protocol for the firewall: Finally, the type of protocols and services (proxies) the firewall will work with must be decided on. The decision is actually based on the type of services that will be offered in the organization network.

12.4 The Demilitarized Zone (DMZ)

A DMZ is a segment of a network or a network between the protected network and the "bad external network." It is also commonly referred to as a service network. The purpose of a DMZ on an organization network is to provide some insulation and extra security to servers that provide the organization services for protocols such as HTTP/ SHTTP, FTP, DNS, and SMTP to the general public. There are different setups for these servers. One such setup is to make these servers actually bastion hosts so that there is a secure access to them from the internal protected network to allow limited access. Although there are restrictions on accesses from the outside network, such restrictions are not as restrained as those from within the protected network. This enables customers from the outside to access the organization's services on the DMZ servers.

Note that all machines in the DMZ area have a great degree of exposure from both external and internal users. Therefore, these machines have the greatest potential for attacks. This implies that these machines must be protected from both external and internal misuse. They are therefore fenced off by firewalls positioned on each side of the DMZ. See Fig. 12.9 for the positioning of DMZ servers.

According to Joseph M. Adams [8], the outer firewall should be a simple screening firewall just to block certain protocols, but let others through that are allowed in the DMZ. For example, it should allow protocols such as FTP, HTTP/ SHTTP, SMTP, and DNS while denying other selected protocols and address signatures. This selective restriction is important not only to machines in the DMZ but also to the internal protected network because once an intruder manages to penetrate the machines in the DMZ, it is that easy to enter the protected internal network. For example, if DMZ servers are not protected, then an intruder can easily penetrate them. The internal firewall, however, should be more restrictive in order to more protect the internal network from outsider intruders. It should deny even access to these protocols from entering the internal network.

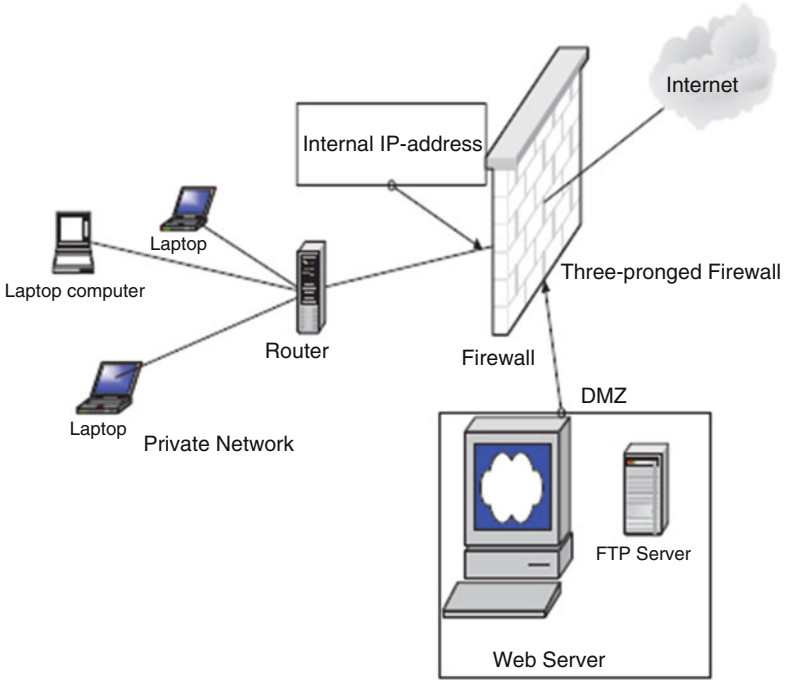

Fig. 12.9 Placing of Web, DNS, FTP, and SMTP servers in the DMZ

Beyond the stated advantage of separating the heavily public accessed servers from the protected network, thus limiting the potential for outside intruders into the network, there are other DMZ advantages. According to Chuck Semeria [9], DMZs offer the following additional advantages to an organization:

- The main advantage for a DMZ is the creation of three layers of protection that segregate the protected network. So in order for an intruder to penetrate the protected network, he or she must crack three separate routers: the outside firewall router, the bastion firewall, and the inside firewall router devices.
- Since the outside router advertises the DMZ network only to the Internet, systems on the Internet do not have routes to the protected private network. This allows the network manager to ensure that the private network is "invisible" and that only selected systems on the DMZ are known to the Internet via routing table and DNS information exchanges.
- Since the inside router advertises the DMZ network only to the private network, systems on the private network do not have direct routes to the Internet. This guarantees that inside users must access the Internet via the proxy services residing on the bastion host.

- Since the DMZ network is a different network from the private network, a network address translation (NAT) can be installed on the bastion host to eliminate the need to renumber or resubnet the private network.

The DMZ also has disadvantages including the following:

- Depending on how much segregation is required, the complexity of DMZ may increase.
- The cost of maintaining a fully functional DMZ can also be high again depending on the number of functionalities and services offered in the DMZ.

12.4.1 Scalability and Increasing Security in a DMZ

Although the DMZ is a restricted access area that is meant to allow outside access to the limited and often selected resources of an organization, DMZ security is still a concern to system administrators. As we pointed out earlier, the penetration of the DMZ may very well result in the penetration of the protected internal network by the intruder, exploiting the trust relationships between the vulnerable host in the DMZ and those in the protected internal network.

According to Marcus Ranum and Matt Curtin [10], the security in the DMZ can be increased and the DMZ scaled by the creation of several "security zones." This can be done by having a number of different networks within the DMZ. Each zone could offer one or more services. For example, one zone could offer services such as mail, news, and host DNS. Another zone could handle the organization's Web needs.

Zoning the DMZ and putting hosts with similar levels of risk on networks linked to these zones in the DMZ helps to minimize the effect of intrusion into the network because if an intruder breaks into the Web server in one zone, he or she may not be able to break into other zones, thus reducing the risks.

12.5 Improving Security Through the Firewall

The firewall shown in Fig. 12.9 is sometimes referred to as a three-pronged firewall or a tri-homed firewall because it connects to three different networks: the external network that connects to the Internet, the DMZ screened subnet, and the internal protected network. Because it is three-pronged, it, therefore, requires three different network cards.

Because three-pronged firewalls use a single device and they use only a single set of rules, they are usually complex. Such a set of rules can be complex and lengthy. In addition, the firewall can be a weak point into the protected network since it provides only a single entry point into two networks: the DMZ network and the internal network. If it is breached, it opens up the internal network. Because of this, it is usually better for added security to use two firewalls as in Fig. 12.10.

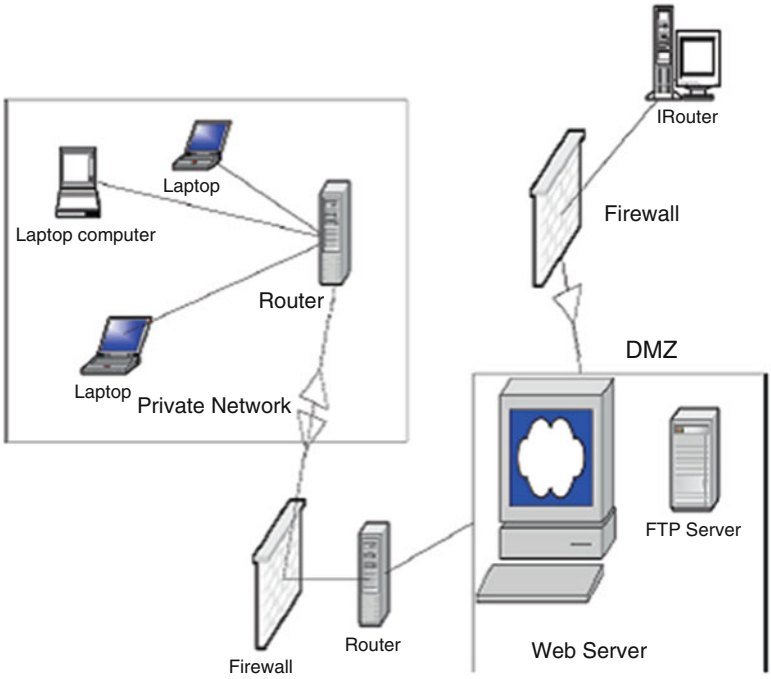

Fig. 12.10 Two firewalls in a network with a DMZ

Other configurations of firewalls depend on the structure of the network. For example, in a setup with multiple networks, several firewalls may be used, one per network. Security in a protected network can further be improved by using encryption in the firewalls. Upon receipt of a request, the firewall encrypts the request and sends it on to the receiving firewall or server which decrypts it and offers the service.

Firewalls can also be equipped with intrusion detection systems (IDS). Many newer firewalls now have IDS software built into them. Some firewalls can be fenced by IDS sensors as shown in Fig. 12.11.

12.6 Firewall Forensics

Since port numbers are one of the keys used by most firewalls, let us start firewall forensics by looking at port numbers. A port number is an integer number between 1 and 65535 which identifies to the server what function a client computer wants to be performed. By port numbering, network hosts are able to distinguish one TCP and UDP service from another at a given IP address. This way, one server machine can provide many different services without conflicts among the incoming and outgoing data.

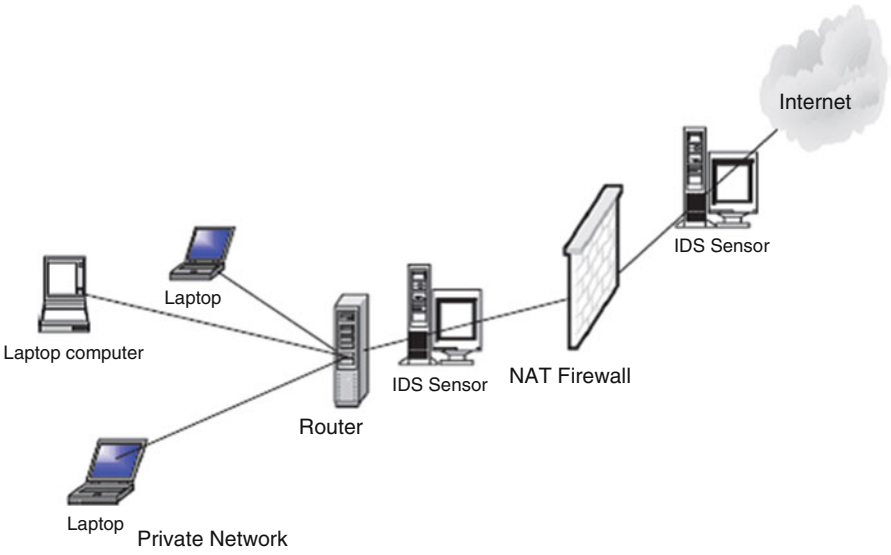

Fig. 12.11 Firewalls with IDS sensors

According to Robert Graham [11], port numbers are divided into three ranges:

- The *well-known ports* are those from 0 through 1023. These are tightly bound to services and usually traffic on these ports clearly indicates the protocol for that service. For example, port 80 virtually always indicates HTTP traffic.
- The *registered ports* are those from 1024 through 49151. These are loosely bound to services, which means that while there are numerous services bound to these ports, these ports are likewise used for many other purposes that have nothing to do with the official server.
- The *dynamic and/or private ports* are those from 49152 through 65535. In theory, no service should be assigned to these ports.

In reality, machines start assigning *dynamic ports* starting at 1024. There is also strangeness, such as Sun starting their RPC ports at 32768 [11].

Using port numbers and in a clear and concise document, Robert Graham explains what many of us see in firewall logs. His document is intended for both security experts and home users of personal firewalls. The full text of the article can be found here: http://www.robertgraham.com/pubs/firewall-seen.html. We encourage the reader to carefully read this document for a full understanding of and putting sense in what a firewalls outputs.

12.7 Firewall Services and Limitations

As technology improves, firewall services have widened far beyond old strict filtering to embrace services that were originally done by internal servers. For example, firewalls can scan for viruses and offer services such as FTP, DNS, and SMTP.

12.7.1 Firewall Services

The broad range of services offered by the firewall are based on the following access controls [4]:

- Service control: Where the firewall may filter traffic on the basis of IP addresses, TCP, UDP, port numbers, and DNS and FTP protocols in addition to providing proxy software that receives and interprets each service request before passing it on.
- Direction control: Where permission for traffic flow is determined from the direction of the requests.
- User control: Where access is granted based on which user is attempting to access the internal protected network, which may also be used on incoming traffic.
- Behavior control: In which access is granted based on how particular services are used, for example, filtering e-mail to eliminate spam.

12.7.2 Limitations of Firewalls

Given all the firewall popularity, firewalls are still taken as just the first line of defense of the protected network because they do not assure total security of the network. Firewalls suffer from limitations, and these limitations and other weaknesses have led to the development of other technologies. In fact, there is talk now that the development of IPSec technology is soon going to make firewall technology obsolete. We may have to wait and see. Some of the current firewall limitations are [11] as follows:

- Firewalls cannot protect against a threat that bypasses it, such as a dial-in using a mobile host.
- Firewalls do not provide data integrity because it is not possible, especially in large networks, to have the firewall examine each and every incoming and outgoing data packet for anything.
- Firewalls cannot ensure data confidentiality because, even though newer firewalls include encryption tools, it is not easy to use these tools. It can only work if the receiver of the packet also has the same firewall.
- Firewalls do not protect against internal threats.
- Firewalls cannot protect against transfer of virus-infected programs or files.

Exercises

1. Discuss the differences between a firewall and a packet filter.
2. Give reasons why firewalls do not give total security.
3. Discuss the advantages of using an application-level firewall over a network-level firewall.
4. Show how data protocols such as TCP, UDP, and ICMP can be implemented in a firewall and give the type of firewall best suited for each of these protocols.
5. What are circuit-level firewalls? How are they different from network-level firewalls?
6. Discuss the limitations of firewalls. How do modern firewalls differ from the old ones in dealing with these limitations?
7. How would you design a firewall that would let Internet-based users upload files to a protected internal network server?
8. Discuss the risks to the protected internal network as a result of a DMZ.
9. What is a bastion router? How different is it from a firewall?
10. Search and discuss as many services and protocols as possible offered by a modern firewall.

Advanced Exercises

1. Many companies now offer either trial or free personal firewalls. Using the following companies, search for a download, and install a personal firewall. *The companies are: Deerfield.com, McAfee, Network Ice, Symantec, Tiny Software, and Zone Labs.*
2. Design a security plan for a small (medium) company and use that plan to configure a firewall. Install the firewall—use some firewalls from #1 above.
3. Zoning the DMZ has resulted in streamlining and improving security in both the DMZ and the protected internal network. Consider how you would zone the DMZ that has servers for the following services and protocols: HTTP/SHTTP, FTP, ICMP, telnet, TCP, UDP, Whois, and finger. Install the clusters in the DMZ.
4. Research the differences between IPSec and firewalls. Why is it that some people are saying that IPSec will soon make firewalls obsolete?
5. Discuss the best ways of protecting an internal network using firewalls from the following attacks:
   - SMTP server hijacking
   - Bugs in operating systems
   - ICMP redirect bombs
   - Denial of service
   - Exploiting bugs in applications

References

1. Kizza JM (2002) Computer network security and cyber ethics. McFarland Publishers, Jefferson
2. Karose J, Ross K (2000) Computer networking: a top-down approach featuring the internet. Addison-Wesley, Boston
3. Holden G (2004) A guide to firewalls and network security: intrusion detection and VPNs. Clifton Paark, Thomson Learning
4. Hall E. Internet Firewall Essentials. http://www.windowsecurity.com/whitepapers/firewalls_and_VPN/Internet_Firewall_Essentials.html
5. Panko RR (2004) Corporate computer and network security. Prentice Hall, Upper Saddle River
6. SANS Institute. Web application firewalls. SANS Institute InfoSec Reading Room. https://www.sans.org/reading-room/whitepapers/application/web-application-firewalls-35817
7. Fennelly C. Building your firewall, Part 1. http://www.windowsecurity.com/whitepapers/firewalls_and_VPN/Building_your_firewall_Part_1.html
8. Adams JM. FTP server security strategy for the DMZ. https://www.giac.org/paper/gsec/805/ftp-server-security-strategy-dmz/101713
9. Semeria C. Internet firewalls and security. A technology overview. http://www.linuxsecurity.com/resource_files/firewalls/nsc/pdf/50061901.pdf
10. Ranum MJ, Curtin M. Internet firewalls: frequently asked questions. http://www.interhack.net/pubs/fwfaq/
11. Graham R. Firewall forensics (What am I seeing?). http://www.lysator.liu.se/~kjell-e/tekla/linux/security/firewall-seen.html