Chapter 3

Designing Firewalls: A Survey

Angelos D. Keromytis and Vassilis Prevelakis

3.1 INTRODUCTION

A *firewall* is a collection of components interposed between two networks that filter traffic between them according to some security policy [1]. Typically, firewalls rely on restrictions in the network topology to perform this filtering. One key assumption under this model is that everyone on the protected network(s) is trusted, since internal traffic is not seen by the firewall and thus cannot be filtered; if that is not the case, then additional, internal firewalls have to be deployed in the internal network. Most of the complexity in using firewalls today lies in managing a large number of firewalls and ensuring they enforce a consistent policy across an organization's network.

The typical firewall configuration, shown in Figure 3.1, usually comprises two packet filtering routers creating a restricted access network called the DMZ (demilitarized zone). The DMZ acts as a buffer between the internal (trusted) and external (untrusted) networks. This configuration attempts to satisfy a number of goals:

- Protect hosts on the internal (inside) network from attacks from the outside
- Allow machines located in the DMZ to be accessed from the outside and thus be able to provide services to the outside world or serve as stepping stones linking hosts from the internal network to the hosts in the outside world
- Enforce an organizationwide security policy, which may include restrictions unrelated to security, for example, access to certain websites during office hours

For a firewall to be effective, it must be strategically placed so that all traffic between the internal network and the outside world passes through it. This implies that firewalls traditionally are located at the points where the internal network is connected to the outside network (e.g., the Internet service provider). These are called the *choke points*. By placing the firewall at the choke points we control all traffic that enters or leaves the internal network. However, as the speed of the network connections increases and the policies that must be applied by firewalls become more complex, firewalls may become bottlenecks restricting the amount of legitimate information that may pass through them.

Network Security: Current Status and Future Directions, Edited by C. Douligeris and D. N. Serpanos

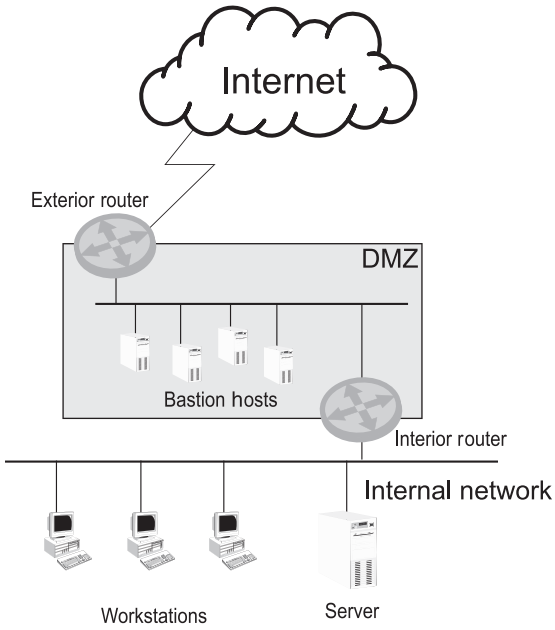

Figure 3.1 Typical firewall configuration.

3.1.1 Demilitarized Zone

The DMZ is a special part of the network that enjoys only partial protection from the firewall. This allows the firewall administrator to establish a special set of policies for these machines. So, for example, while the main security policy may dictate that internal hosts may not be contacted from the outside network, a special DMZ policy may allow exceptions so that a Web server located in the DMZ may be contacted over the Transmission Control Protocol (TCP) port 80 or so that the e-mail server may be contacted over the SMTP (Simple Mail Transfer Protocol) port, TCP port 25.

The positioning of the hosts in the DMZ also makes them more vulnerable, which is why they are usually configured with special attention to their security. Such hosts are sometimes referred to as *bastion hosts*. Bastion hosts, while they are general-purpose computers running a general-purpose operating system, usually have highly specialized configurations allowing them to run only the designated services and nothing more. Sometimes, these machines run with statically assigned operational parameters [e.g., using the /etc/hosts file for name resolution rather than the domain name system (DNS)]. This is so as to minimize the risk that an attacker may use a service unrelated to the function of the machine to gain a foothold. Moreover, the software installed on bastion hosts is a subset of the standard distribution (e.g., may lack compilers, network monitoring tools, etc.) so that a potential intruder will not be able to use the compromised machine to launch attacks on other machines in the network.

Administrators must assume that eventually hosts in the DMZ will be compromised and establish recovery strategies. Such strategies may include steps to contain the attack, to gather evidence of the break-in or information about the attacker, and so on. Regardless of the adopted strategy, the system administrator must be able to restore service on the compromised machine as soon as possible. This implies that the entire configuration of

the machine has been backed up and procedures exist for the reinitialization of the infected machine and the restoration of its configuration and associated data sets. Unless the method used by the attacker is identified, merely bringing the machine back online with a clean configuration is not enough. The attacker will simply use the same attack vector to compromise the machine once again. We need to identify the vulnerability that allowed the attack to take place and fix it before the machine can be connected to the network. Detecting and understanding the attacks that take place against hosts in the DMZ or perhaps the internal network are important aspects of a firewall. Traffic monitoring and event logging are the primary tools of the network administrator. Intrusion detection systems (IDSs) may also be installed in the DMZ to detect and sometimes respond to attacks.

3.1.2 Packet Filters Versus Application-Level Gateways

The two routers in the example above employ some rules (e.g., an access control list) to determine which types of packets to allow through. Packet-level filtering is rather coarse as it is positioned at the network and transport layers and hence has little or no information about what is happening at the application level. Thus, policies such as *"only user X may access www.cnn.com over HTTP during working hours"* cannot be expressed.

Higher level policies that require specific knowledge of the application (e.g., e-mail virus scanners) or user authentication are best handled by proxy servers, also known as application-level gateways. Such machines typically are located in the DMZ and process traffic for specific applications.

One such example is the e-mail gateway. Typically, the e-mail server is located in the protected network as it has to deal with internal e-mail as well. In order to prevent a compromise of the e-mail server, we do not want to allow it to accept direct connections from the outside network (Internet). We therefore position an e-mail proxy in the DMZ which simply collects inbound e-mail. The e-mail server then contacts the proxy at regular intervals to pick up any e-mail that may have arrived in the meantime. Notice that the e-mail proxy is totally passive; it is waiting to be contacted by the internal e-mail server or by outside hosts. This ensures that even if the proxy were to be compromised, the intruder would not be able to probe or attack the internal server.

Of course, this arrangement can only protect against network attacks; it cannot protect from data bombs such as viruses. Additional analysis has to be carried out of the contents of the e-mail messages in order to determine whether they contain suspicious content. To do this, the gateway needs to understand the way e-mail messages are constructed (i.e., encoding standards such as Multipurpose Internet Mail Extension (MIME), uuencode, zip, etc.). Since attackers constantly come up with different strategies, the defenders need to be very rigorous in keeping up with security advisories and virus signatures. This increasingly looks like a full-time task, and often companies subcontract the analysis of inbound e-mail to outside security firms. In such cases, e-mail may be diverted over the Internet to the site of a security firm where it is analyzed and evaluated. E-mail that is considered safe is then returned to the e-mail proxy where it may be picked up by the internal server.

3.1.3 Stateful Firewalls

Originally, firewalls were designed to deal with each packet individually, forcing the firewall to determine whether to allow a packet through only on the basis of the information contained within that packet.

This created difficulties with protocols that relied on secondary connections for the exchange of additional information [e.g., File Transfer Protocol (FTP)]. Since the firewall could not know whether the (secondary) connection request was issued by an existing connection or it was created independently, the firewall was forced to reject it.

Stateful firewalls employ state machines to maintain state associated with established protocol connections. Decisions are made on the basis of the information in the packet plus the state of the connection maintained by the firewall. Thus, a TCP packet with the SYN flag cleared will be rejected unless it belongs to an already established connection.

Even in cases where information is exchanged without setting up a connection [connectionless communications such as those carried over the User Datagram Protocol (UDP)], the firewall can make a note that a request packet has passed on its way out of the protected network and thus allow the reply through [e.g., a Simple Network Management Protocol (SNMP) query from an internal network management station to an agent located in the DMZ].

3.1.4 Additional Services

In many situations, firewalls also provide a number of additional services which, while not strictly part of the firewall "job description," have been used so widely that they are now considered an integral part of a firewall.

3.1.4.1 Network Address Translation

The ever-increasing scarcity of Internet Protocol (IP) addresses has been forcing network administrators to use special IP addresses that are considered private. Such addresses may be used only within the boundaries of a given network but are meaningless on the Internet. This is because they are not unique, so the backbone routers carry no routing information about them.

If hosts with private IP addresses require access to the Internet, they must use an intermediary host that has a global address. Such a host may act as a proxy, relaying the request to the final destination.

However, proxies may not always be usable because of limitations of the protocol, the use of end-to-end encryption, but, most importantly, the additional administrative cost of setting up and maintaining separate proxies for each of the desired services. In such cases the use of network address translation (NAT, or IP masquerade) is recommended. Under a NAT regimen the intermediary host modifies the outgoing packet changing the source address to its own address. In this way, the response will be received by the intermediary host which will again modify the packet's destination address to that of the internal host. Given the location of firewall assets in the network, its is quite natural to assign the NAT task to them. This is because firewalls already have to examine (for packet filtering purposes) packets that cross the network boundaries and also because firewalls already maintain state about the connections that exist between internal and external hosts.

3.1.4.2 Split-Horizon DNS

The DNS provides information related to the mapping between IP addresses and hostnames. This information may be used by an attacker to identify targets (e.g., a machine

called mailhost is likely to be the mail server of the organization and hence have mail-related services activated). For this reason two DNS servers are often employed, one for the internal network and one on the DMZ providing information to outside hosts. The internal DNS server maintains information about all hosts in the internal network, while the server in the DMZ stores only information that should be known to outside parties (generally names of machines that are accessible from the outside).

3.1.4.3 Mitigating Host Fingerprinting

Computer systems are to a large extent deterministic, and this can be used as a means of identification (fingerprinting) or, worse, as a means of subverting a system by anticipating its response to various events.

Fingerprinting is a technique that allows remote attackers to gather enough information about a system so that they can determine its type and software configuration (version of operating system, applications, etc.). This information can then be used to determine what vulnerabilities may be present in that configuration and thus better plan an attack.

Many packet filtering firewalls include a "scrub" function that normalizes and defragments incoming packets. This allows applications and hosts on the internal network some protection against hand-crafted packets designed to trigger vulnerabilities. Another approach is to apply a similar technique to outgoing packets in order to hide identifying features of the IP stack implementation.[1] A key part of the obfuscation process is protection against time-dependent probes. Different TCP implementations have variations in their timeout counters, congestion avoidance algorithms, and so on. By monitoring the response of the host under inspection to simulated packet loss, the timing probe can determine the version of the TCP implementation and by extension that of the operating system (OS). Also the use of various techniques for rate-limiting Internet Control Message Protocol (ICMP) messages by the victim system can provide hints to the attacker. The effectiveness of such probes can be reduced by homogenizing the rate of ICMP traffic going through the firewall or by introducing random delays to ICMP replies.

3.1.4.4 Intrusion Detection Systems

A corollary of the "there is no perfect security" rule is that your firewall assets will be eventually compromised. With this in mind, it is imperative to have a strategy for detecting and responding to the security breach. Intrusion detection systems (IDSs) are naturally placed within the DMZ and may be traffic monitors or booby-trapped hosts. Traffic monitoring systems tap into all traffic that crosses the DMZ and attempt to identify patterns that may indicate an attack. Booby-trapped systems (also known as *honeypots*) are systems that are configured to look like potential targets for attack (e.g., running many services, running old versions of software that are known to contain vulnerabilities, etc.). Since authorized users of the network know that they should not be using the honeypot host, anybody who does try to access this host is, by definition, an intruder.

Output from the IDS is used as a signal to trigger attack containment and mitigation actions that are described later in this chapter. IDSs are discussed in greater detail in Chapter 6.

[1] http://www.insecure.org/nmap/nmap-fingerprinting-article.html.

3.1.5 Limitations of Firewalls

Firewalls are widely considered to be necessary as general-purpose computers are difficult to protect. Nevertheless, a mythical "general-purpose firewall" would be essentially useless. In order to be effective, firewalls need to be customized to the needs of their environment. For example, home firewalls generally block incoming connections, but if the home owner wishes to set up a website to be able to receive e-mail, then the firewall would have to be reconfigured.

Despite the advances made in the past 10 years, firewall configuration is still a difficult and error-prone procedure, requiring careful verification and testing to ensure that the firewall does exactly what we want. In order to do this, the administrator needs to understand the requirements of the network that will be protected by the firewall, the requirements and the protocols used by the various applications that should be allowed through the firewall, and, finally, the way the firewall itself enforces the configuration defined by the administrator.

Subtle differences between what we expect the firewall to do and what it actually does may cause difficulties with the operation of authorized applications or, perhaps, allow unauthorized traffic through the firewall.

The "short-packet" attack is a good example of a situation where the attacker tries to force the firewall to make a decision with insufficient data. This attack relies on the observation that since many firewalls do not reassemble fragmented packets they must base their decision on the first fragment of the packet and allow the rest through, essentially unchecked. The short-packet attack fragments packets so that the first fragment does not contain the entire TCP header (and thus lacks information such as the destination port). Modern firewalls typically reject such packets.

Other limitations of traditional firewalls include the following:

- Due to the increasing line speeds and the more computationally intensive protocols that a firewall must support, firewalls tend to become congestion points. This gap between processing and networking speeds is likely to increase, at least for the foreseeable future: While computers (and hence firewalls) are becoming faster (following Moore's law), protocols and the tremendous increase in the amount of data that must be processed by the firewall have been and will likely continue to outpace Moore's law [2].
- The increasing scale of modern networks typically implies a large number of attachments to the Internet for performance, fault tolerance, and other reasons. Firewalls need to be deployed on all these links, greatly increasing the management problem.
- The increased scale also means that often there are attackers already on the inside network, for example, a disgruntled employee. Traditional firewalls can do very little, if anything, against such a threat.
- Furthermore, the use of wireless (802.11 or similar) networks, whether authorized or not,[2] means that administrators do not necessarily have tight control on the network entry points: Attackers or free-loaders can appear from inside the network. Similar concerns arise due to the increased use of telecommuting facilities, which

[2] For example, consider the case of a user who simply connects a wireless base station on the corporate local area network (LAN) so that he can work from the corporate lounge.

de facto extend the boundary of the protected network to include infrastructure resident in, for example, employees' premises. While firewalls are generally not intended to guard against misbehavior by insiders, there is a tension between internal needs for more connectivity and the difficulty of satisfying such needs with a centralized firewall.

- End-to-end encryption can also be a threat to firewalls, as it prevents them from looking at the packet fields necessary to do filtering. Allowing end-to-end encryption through a firewall implies considerable trust to the users on behalf of the administrators.
- There are protocols that firewalls find relatively difficult to handle because they involve multiple, seemingly independent packet flows. One example is FTP, where a control connection is initiated by the client to the server but (at least in some configurations) data connections are initiated by the server to the client. Although modern firewalls can and do handle these protocols, such solutions are viewed as architecturally "unclean" and in some cases too invasive.
- Finally, there is an increasing need for finer grained (and even application-specific) access control which standard firewalls cannot readily accommodate without greatly increasing their complexity and processing requirements.

Despite their shortcomings, firewalls are still useful in providing some measure of security. The key reason that firewalls are still useful is that they provide an obvious, mostly hassle-free, mechanism for *enforcing* network security policy. For legacy applications and networks, they are the only mechanism for security. While newer protocols sometimes have some provisions for security, older protocols (and their implementations) are more difficult, often impossible, to secure. Furthermore, firewalls provide a convenient first-level barrier that allows quick responses to newly discovered bugs.

3.2 FIREWALL CLASSIFICATION

Apart from the typical firewall configuration described in the introduction to this chapter, there exist a number of other firewalls that are customized for particular applications or environments. In this section we examine some of the most popular configurations.

3.2.1 Personal Firewall

The term *personal firewall* generally refers to software that runs on your workstation and acts as a packet filtering firewall. The advantage of the personal firewall is that it can associate rules with programs so that, for example, your Web browser can connect to hosts all over the Internet over the HyperText Transfer Protocol (HTTP) port (port 80), but your word processor cannot. This works because the firewall is located on the same machine as the process that sends the packets. The personal firewall installs kernel-level software that monitors and intercepts network-related calls. In this way the firewall can determine which process is sending the packets.

Nevertheless, the concept of the personal firewall has a number of weaknesses. Namely, it runs under a general-purpose operating system and must coexist with services that run with elevated privileges (sometimes without the user even being aware of it). If a privileged process is compromised, then the firewall can be confused or even subverted.

Lately, one of the first actions of viruses that take over machines is to turn off the virus checking software. It is only a matter of time before they start disabling the personal firewall on that machine.

Another major limitation is based on the fact that the trust associated with a process is inherited by its children. So while a virus cannot make a process perform actions that are not part of its authorized execution profile, it can take advantage of all the privileges enjoyed by that process. Thus, assuming that network-aware processes can be infected, the intruder will have all the privileges of the infected process, which may be more than adequate to carry out its mission.

One such exploit that runs under the Windows operating system has recently been described in great detail by Rattle [3].

3.2.2 Distributed Firewall

Conventional firewalls rely on topology restrictions and controlled network entry points to enforce traffic filtering. Furthermore, a firewall cannot filter traffic it does not see, so, effectively, everyone on the protected side is trusted. While this model has worked well for small- to medium-size networks, networking trends such as increased connectivity, higher line speeds, Extranets, and telecommuting threaten to make it obsolete.

To address the shortcomings of firewalls while retaining their advantages, [4] proposed the concept of a *distributed firewall.* In distributed firewalls, security policy is defined centrally but enforced at each individual network endpoint (hosts, routers, etc.). The system propagates the central policy to all endpoints. Policy distribution may take various forms. For example, it may be pushed directly to the end systems that have to enforce it, or it may be provided to the users in the form of credentials that they use when trying to communicate with the hosts, or it may be a combination of both. The extent of mutual trust between endpoints is specified by the policy.

To implement a distributed firewall, three components are necessary:

- A language for expressing policies and resolving requests. In their simplest form, policies in a distributed firewall are functionally equivalent to packet filtering rules. However, it is desirable to use an extensible system (so other types of applications and security checks can be specified and enforced in the future). The language and resolution mechanism may also support credentials for delegation of rights and authentication purposes [5].
- A mechanism for safely distributing security policies. The integrity of the policies transfered must be guaranteed, either through the communication protocol or as part of the policy object description (e.g., they may be digitally signed).
- A mechanism that applies the security policy to incoming packets or connections, providing the enforcement part.

3.2.3 Layer 2 Firewall

As we have seen in the earlier sections, firewalls typically operate at the internetwork (IP) layer. This is mainly due to the placement of most firewalls: They usually replace the traditional router that connects the internal network with the external untrusted network. Thus, the firewalls were designed to operate at the same layer as the machine that they replaced (the routers).

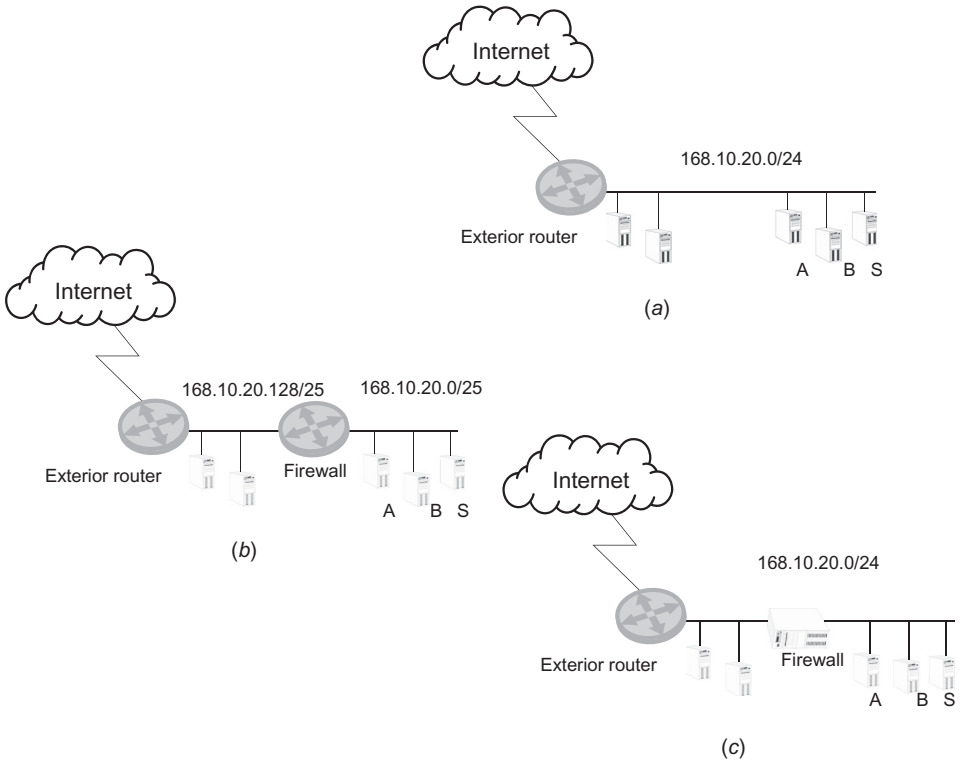

Figure 3.2 Layer 2 firewall. (*a*) Network without internal firewall requires a single IP address space. (*b*) Adding a firewall as a router requires the network address space to be split in two. (*c*) Adding a bridging firewall can be done without any modifications to the network or hosts.

However, there are cases where we would like to position our firewall as a "bump in the wire," that is, so that it is transparent to the rest of the network elements. Achieving this while operating at the IP layer is difficult as it would necessitate the creation of a new network between the firewall and the external router (see Fig. 3.2) [4].

The transparency of the layer 2 firewall to the IP hosts allows the insertion of a firewall without disrupting the operation of the network. In fact, the various hosts and related network elements need not be aware of the installation of the firewall. This feature of the layer 2 firewall allows easy deployment (essentially on demand) in order to provide increased security to a specific segment of the internal network, to troubleshoot a problem, or to mitigate an ongoing attack (e.g., if some hosts are infected by a new virus, layer 2 firewalls can be deployed at various points in the network to prevent the spread of the infection).

3.2.3.1 Example of Use of Layer 2 Firewall

Assume that we have a number of hosts located on the same network and we would like to allow some services from host S to be available to hosts A and B (Fig. 3.2*a*) but not to the other hosts in the network. We could create a small net comprising hosts S, A, and B and link it to the main network with a firewall F. However, in this case we would need to come up with new addresses for hosts S, A, and B, which were outside the main network.

We would then have to make sure that routing changes were instituted throughout the main LAN to ensure that packets for S, A, and B were sent to F. If addresses for the new network were not available, then F would have to perform some additional modifications to the packets (e.g., network address translation), further complicating the firewall configuration.

Using a layer 2 firewall, the three hosts (S, A, and B) are placed in a separate Ethernet LAN with the firewall (F) acting as a bridge between the new LAN and the main LAN (Fig. 3.2*c*). Since bridging is done at the Ethernet layer, it is transparent to the IP layer, thus allowing the hosts to retain their original IP addresses for the main network. Thus the firewall may be installed without any kind of modification to the hosts [even services such as DHCP (Dynamic Host Configuration Protocol) will be unaffected]. The firewall may then block access to the restricted services to all hosts on the main network.

3.2.3.2 Using Layer 2 Firewall to Prevent ARP Spoofing Attacks

A host that wishes to send a packet to another host on the same network needs to locate the Ethernet [or media access control (MAC)] address of the recipient machine. It must, thus, find out which MAC address corresponds to the IP address of the recipient. Under IP version 4, hosts use the Address Resolution Protocol (ARP) to perform this conversion. The ARP requires that the sending host broadcast an Ethernet packet containing the recipient's IP address essentially asking who has that IP address. The owner of the IP address will then reply directly to the host that made the inquiry. In some cases hosts such as routers may send ARP packets with their IP and MAC addresses to prevent hosts from clearing these mappings from their caches. Such transmissions are called *gratuitous ARPs*.

ARP spoofing attacks typically involve a (hostile) host (H) that issues fake gratuitous ARP packets providing *its* MAC address for the address of a host (R) that is to be spoofed. If the recipient (S) of the gratuitous ARP packet has the IP address in its cache, it will replace the corresponding MAC address with the new (spoofed) MAC address. In a switched (or bridged) Ethernet LAN the real owner of the IP address will not detect the spurious activity because the transmission is unicast.

The victim host will now send all packets destined for R to H because its ARP cache has been contaminated. Host H can now either passively monitor the transmissions of host S or engage in an active man-in-the-middle attack by modifying the packets that flow through it.

ARP spoofing attacks are particularly effective when used to spoof the local default router or the DNS server and are quite difficult to detect.

Assuming the configuration used in our previous example, firewall F will allow ARP packets through while verifying that the information within them is consistent with previous traffic and flag cases where MAC-to-IP address mappings change.

Despite their benefits, the use of layer 2 firewalls is rather limited because of concerns about their efficiency and administrative overheads. Filtering Ethernet frames is considered more resource intensive, creating fears that layer 2 firewalls may not be able to keep up with the traffic generated by modern high-speed LANs. Also the added complexity imposed by the need to create rules that operate at the Ethernet layer has created the impression that layer 2 firewalls are more difficult to configure. Justified or not these two criticisms have generally kept layer 2 firewalls from corporate networks.

3.2.4 Appliance Firewall

Both the distributed and the personal firewalls have the disadvantage that they are running on the same hardware (and under the same general-purpose operating system) as user-level applications. As a consequence, any breach of security by one of the other applications (e.g., a virus infection) may interfere with the operation of the firewall.

Because of limitations in the design of most of the current popular operating systems, personal firewalls are likely to provide only a false sense of security, rather than actual protection. In the case of the distributed firewall, policy enforcement mechanisms operating at the system call level provide additional protection. Nevertheless, the operation of the firewall may be affected by user actions (intentional, accidental, or induced by an attacker using human engineering).

Such concerns are addressed by the appliance firewall, which is a dedicated hardware device external to the host that we want to protect. The appliance firewall generally acts as a traditional firewall, but it only protects a single host. The appliance firewall has two interfaces, one to connect to the computer it protects and another that connects to the rest of the network. The host always communicates with the outside world via the appliance firewall.

Since the appliance firewall must implement the site security policy, there is a need for distributing this policy to all appliance firewalls in the network in a secure manner. This may be achieved in two ways: (a) have the appliance firewalls download security policy updates at regular intervals (this is similar to the automatic downloading of virus signature files) or (b) the user of the protected host initiates a policy update (for example, so that he can perform a new task that is not covered by existing policy) [7].

Appliance firewalls are particularly effective in helping mobile users secure their laptops. Under this scenario, the appliance firewall may be used as a virtual private network (VPN) gateway to allow the mobile user access to the home network. For appliance firewalls to be effective in these diverse roles, they must be easy to use and inconspicuous. As can be seen in Figure 3.3, the latest generation of appliance firewalls have shrunk to the point where they pose little burden to the mobile user.

3.3 FIREWALL DEPLOYMENT: MANAGEMENT

3.3.1 Location

As we have already discussed, traditional firewalls exploit restrictions in the network topology to enforce a security policy. What worked well in the early days of networking, however, where most organizations had relatively small networks with one or at most two connections to a public network, does not necessarily scale in today's environments. As a result, considerable care must be taken in determining placement of firewall assets.

Organizations still try, as much as possible, to follow the perimeter firewall model, where one firewall sees all traffic to and from that organization's network and enforces its security policy. The primary reason for this is manageability—the administrator only needs to reconfigure a small number of boxes to effect a change in the security policy. Ensuring the physical integrity of the firewall is also easier when it is composed of only a few systems.

Other benefits of such centralized placement are due to the traffic aggregation seen "deeper" in the network infrastructure (as opposed to the edges). Large-scale phenomena,

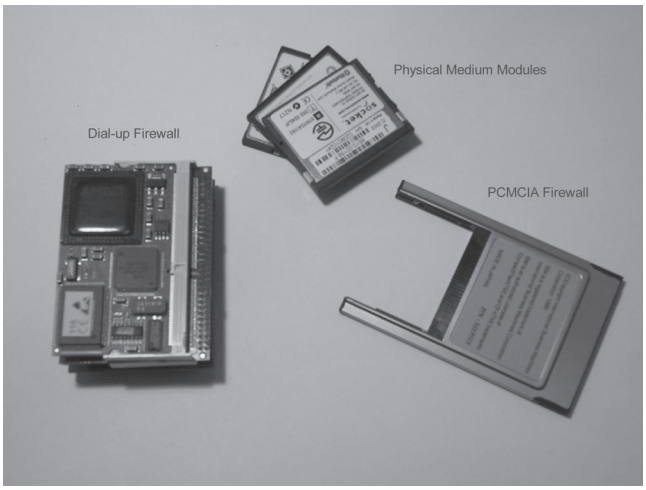

Figure 3.3 Two types of appliance firewalls: The larger one (on the left) is designed for dial-up use, while the one on the right contains the firewall computer on the adapter card, using the daughter boards to provide compatibility with a number of physical media, such as wired and wireless Ethernet, Bluetooth, etc.

such as worm outbreaks, denial-of-service (DoS) attacks, or enterprisewide port scanning and fingerprinting, are easier to detect if all the organization's traffic is seen by the same IDS. Likewise, countering some of these events can only be done in the network core: Filtering a DoS attack at the targeted host is almost worthless, since the damage (clogging the network links) has already been done.

In reality, several such perimeter firewalls are often used, as shown in Figure 3.4:

- For redundancy (fail-over) reasons, a small pool of firewalls share the burden of managing one network uplink. Several commercial firewalls allow state sharing between members of this cluster to ensure transparent operation in the event of failure of any one member.
- The cluster approach also serves to mitigate the performance impact of firewalls by load balancing traffic across its members, typically on a per-session basis, that is, all packets belonging to the same TCP connection, all packets originating from or destined to the same host, and so on. Load balancing becomes imperative when more heavyweight functionality is operated at the firewall, such as application-level monitoring and filtering, VPN functionality (which we cover next), spam/virus scanning, and so on. Tuning firewall performance remains a "black art," often performed by the administrator during system operation.
- Typical organizations have multiple connections to the public network (Internet) nowadays, often for fail-over reasons. Furthermore, different branches of an enterprise are likely to have their own, local network connections, requiring their own firewall (or firewall cluster).

Modern organizations further augment their perimeter firewalls with auxiliary, internal firewalls that protect specific networks and resources. This partitioning of the internal network is often done across departmental boundaries and mirrors the "need-to-know" (or "need-to-access") approach to security. For example, the legal and financial departments are likely to have their own firewalls, since they manage sensitive information that needs

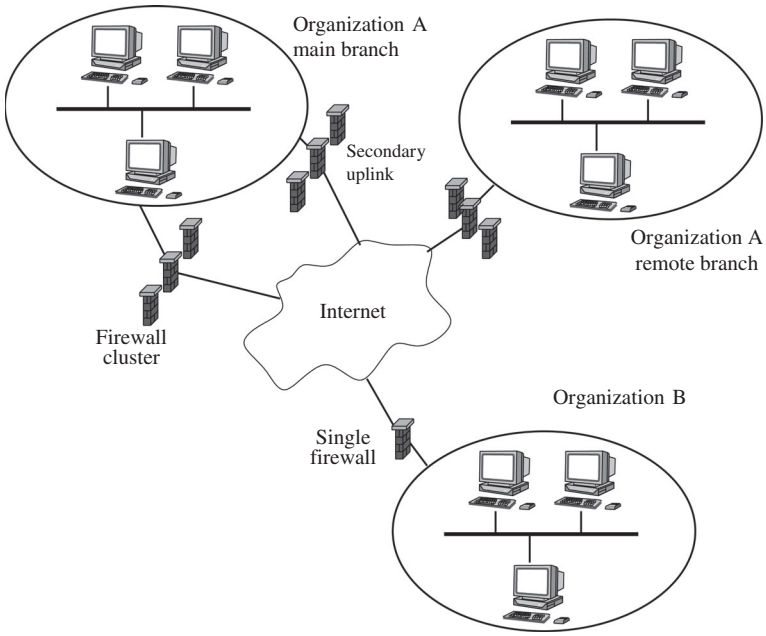

Figure 3.4 Organization A has two geographically (and topologically) distinct branch network, each with its own uplink to the Internet. Organization A's main branch also has a secondary uplink and uses clusters of firewalls for redundancy and performance reasons. Organization B has only one network attachment and uses a single firewall.

to be protected from other company employees as much as from outsiders. Such auxiliary firewalls also serve as a secondary barrier against outside attackers that somehow manage to penetrate into the organization's internal network.

Internal firewalls are also used to define the boundaries of so-called Extranets. These are simply virtual networks constructed on top of physical resources (network links, routers, servers) contributed by two or more cooperating organizations. This is often done to facilitate information exchange and collaboration on specific projects. The role of firewalls placed "around" the physical resources contributing to an Extranet is to prevent external users who are legitimate Extranet participants from gaining access to other resources that happen to be topologically close but administratively distinct from the Extranet.

Finally, firewalls are often used to mediate access between the increasingly common local area wireless networks, such as 802.11 WiFi, and the rest of the enterprise network. Many organizations treat their wireless infrastructure as part of the public network, requiring users to log in to the firewall before being admitted to the internal network even when wireless security features (such as encryption and authentication) are enabled.

From a technical standpoint, there is no difference between internal and perimeter firewalls. It is often the case that the latter are faster and more expensive, since they need to handle significantly more traffic, although that need not always be the case. Intrusion prevention functionality, which we discuss in Section 3.3.3, is more often used by internal firewalls. Especially as it relates to quarantining subnets or hosts in the event of a worm infection, internal firewalls allow the quick containment of such systems before the worm can spread to the rest of the organization.

3.3.2 Virtual Private Networks

Firewalls are the natural endpoints for secure links that often comprise VPNs.[3] The reason for not allowing VPNs to go over firewalls is that if the information carried over the VPN is encrypted, then the firewall will not be able to apply the network security policy to it.

Moreover, some VPN implementations (e.g., those employing IPSEC protocols) are incompatible with NAT (see Section 3.2) and thus the VPN cannot extend to internal hosts with private IP addresses.

In any case, VPN implementations must include a packet filtering firewall to determine which packets will get sent through the VPN. In order to prevent spoofing or injection attacks, the VPN firewall must also examine the incoming packets: If they are coming from outside the VPN but appear to belong to hosts that are part of the VPN, then the firewall will reject them, because they are spoofed. In general, we have three possible responses to packets:

- They should be sent via the VPN.
- They should be sent outside the VPN (i.e., in the clear).
- They should not be sent at all.

Such decisions are crucial to the security of the VPN because they determine the enforcement of the separation between the VPN and the (potentially untrusted) network that carries the VPN traffic. For example, let us assume that Alice, a sales manager of a large corporation, visits some clients. Since she will need to connect to the home network, she has VPN client software installed on her laptop. The VPN configuration must determine what happens if Alice needs to connect to a site on the public Internet. The corporate policy may require that Alice must always go through the corporate network, in which case the VPN software on her laptop will direct all outgoing packets to the VPN. Once these packets reach Alice's home network, they will be sent again to the Internet (this time unencrypted) and the response will be sent via the VPN to Alice. Thus, packets will cross the Internet twice, once via the VPN and another time in the clear. Of course, if the VPN becomes somehow inoperable, Alice will not be able to connect to any host on the Internet.

Another configuration may allow packets that are destined for hosts outside the VPN to bypass the VPN and be sent directly to their final destination. This configuration will allow Alice to communicate with hosts that are not part of the VPN without the need for the redundant round trip to company headquarters. However, this approach may allow malicious content to be deposited on Alice's laptop.

Thus, the chief security concern with VPN clients is what happens to them while they are away from the home base. If these are connected to other networks, they may be infected by viruses or even be used as stepping stones in an attack against the internal network. Even with the earlier scenario where Alice's laptop always goes through the VPN, malicious content may still get through, via nonnetwork means (e.g., USB memory device, CDROM, Data DVD, and so on). For these reasons, VPN connections from the outside are not fully trusted and external users are forced to use DMZ-style networks that provide limited services.

[3] VPNs are discussed in Chapter 4.

3.3.3 Damage Mitigation Techniques

From our discussion so far, it should be obvious that firewalls act primarily as *damage prevention* mechanisms. Their primary role is to keep unauthorized entities outside the protected network by enforcing the organization's security policy. Often, however, the policy or the mechanisms that enforce it prove to be incapable of warding off an attack. In that case, administrators are expected to manually intervene, often alerted by an IDS that detects a specific attack or a general anomaly (e.g., the arrival of too many short UDP packets).

Since administrators are not always available, and as the tempo of some attacks makes reaction at human time scales infeasible, modern firewalls increasingly employ automated countermeasures. Some of these include intrusion prevention and quarantining.

3.3.3.1 Intrusion Prevention Systems

Since administrators often react to attacks after being alerted by an IDS,[4] it makes sense to tie together access control and intrusion detection functionality. In principle, this can allow firewalls to react quickly to improper behavior from otherwise legitimate users (e.g., an attack from a malicious insider or from a telecommuter's system that has been compromised). Intrusion Prevention Systems (IPSs) can also allow for somewhat more permissive treatment of outside or unknown users by allowing them to interact with protected systems in limited ways; if an attack (or suspicious behavior) is detected, these privileges can be automatically revoked.

In practice, IPSs are only as good as the IDSs that control them. A common problem of IDSs is the amount of false positives they generate, that is, the number of times they misidentify legitimate behavior as suspicious. Frequent reconfigurations can cause significant performance degradation and even loss of functionality, for example, by exhausting the firewall's policy tables with bogus rules.

Furthermore, an adversary that is aware of the IPS can "game" the system, often toward mounting a DOS attack against a legitimate user or the entire organization. For example, by sending spoofed packets purporting to arrive from a legitimate telecommuting user, it is often possible to prevent that user from accessing the internal network. Such an attack may otherwise have been impossible for the attacker.

From the organizations point of view, most IDSs also exhibit an unacceptable number of false negatives, that is, they misidentify attacks as legitimate behavior (and do not raise an alert). Depending on the particular system, false-negative rates can be significantly lower than 1%. In the current environment, however, attacks can be launched repeatedly from different locations with impunity. Since the cost of a successful attack to the organization may be prohibitively high (e.g., loss of financial or product development data), it is unwise to depend on an IDS as the only line of defense. Thus, IPSs are often used to detect misbehaving of legitimate users, with outsiders being governed solely by access control rules.

3.3.3.2 Host–Subnet Quarantining

With the drastic increase of network worm and virus outbreaks in recent years, organizations have turned to firewalls as a means of containing such attacks. The first, obvious

[4] IDSs are discussed in Chapter 6.

step is to update the perimeter firewall's policy to contain newly discovered attacks. This represents simply a change in the tempo of rereconfiguration and is by itself insufficient to counter the threat of worms. These can often appear without prior warning ("zero-day" worms) or manifest on the inside of the organization's network without being noticed by the firewall. This is possible by the use of encryption [e.g., a user receiving an encrypted e-mail or accessing an infected Web server over secure sockets layer (SSL) connection] and user mobility (e.g., a user bringing an already infected laptop inside the organization's network).

Thus, internal firewalls are increasingly used to quarantine subnets or specific hosts that exhibit suspicious behavior by taking advantage of some of the observable characteristics of fast-spreading worms. For example, worms such as Slammer [8] or CodeRed [9] send a large number of packets to different hosts over a short period of time. Likewise, most e-mail worms use their own SMTP engine, directly contacting remote servers (as opposed to sending e-mail messages through the organization's servers). Other types of attacks, such as DOS, also generate large volumes of traffic, often using spoofed source IP addresses.

Internal firewalls, often deployed at the LAN level, can block off hosts that appear to have been infected (or otherwise participate in an attack). The simplest way of doing so is to filter all traffic from that host/subnet, disable the port on the Ethernet switch whence the traffic originates, or disassociate the host from the access point (and prevent it from associating again) in wireless networks. In the more advanced quarantining approaches, the infected host is placed in a virtual LAN (VLAN) that allows it to access a Web server containing the latest software patches for several operating systems. The user can then install these patches and restart the system without the worm or danger of being reinfected.

This approach is also used proactively: When a new node appears on the network, the firewall scans it for known vulnerabilities using the same techniques (and often the same software) that attackers use to identify vulnerable hosts. If the firewall determines that the host is running software that is known to be vulnerable and has not been patched, the host is placed in the same VLAN and the user directed to a Web page with instructions on how to update the system. All hosts that attach to the network are scanned at first; often, the firewall will periodically rescan all nodes to detect vulnerable services that were started after the initial (or previous) scan. In some environments, known users that authenticate to the network (as opposed to guests) are spared this scanning but are subject to quarantining if the IPS detects an infestation.

3.4 CONCLUSIONS

We have discussed the concept of the network firewall, from its initial form as a device residing at the perimeter of an organization's network to its current near ubiquitousness in the form of internal (partitioning), distributed, personal, and layer 2 firewalls, as well as the use of firewall clusters for redundancy and performance. In all its guises, a firewall remains a means for administrators to enforce consistently an organizationwide policy on all network traffic entering or leaving the organization's network (and, in the case of internal firewalls, traffic crossing the internal partitions).

Distribution of enforcement functionality allows more flexibility in defining security policies that accurately map the needs of the organization. At the same time, however, the complexity of managing such security policies increases considerably. The increasing use

of wireless networks that topologically reside inside an organization's security perimeter further complicates management.

Current trends in firewall design include the use of multiple firewalls at various locations at the perimeter and inside a network, extensive use of VPN capabilities to form Intranets and Extranets, integration of intrusion detection and prevention functionality (automating the reaction to anomalous events), and use of quarantining mechanisms for containing DOS attacks and virus infestations.

Although considerable research and development have been devoted in extending the capabilities of firewalls [1, 4, 10–21], we predict further developments and refinements of the basic concept as well as increased deployment and use.

REFERENCES

1. W. R. Cheswick and S. M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley, Reading, MA, 1994.
2. M. Dahlin, Serverless Network File Systems, PhD thesis, University of California, Berkeley, Dec. 1995.
3. Rattle, Using process infection to bypass windows software firewalls. *Phrack*, 13(62), July 2004.
4. S. M. Bellovin, Distributed firewalls, *login: magazine, special issue on security*, Nov. 1999, pp. 37–39.
5. M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, The role of trust management in distributed systems security, in *Secure Internet Programming*, LNCS 1603, Springer-Verlag, New York, 1999, pp. 185–210.
6. T. A. Limoncelli, Tricks you can do if your firewall is a bridge, in *Proceedings of the first USENIX Conference on Network Administration*, Santa Clara, CA, Apr. 1999.
7. V. Prevelakis and A. D. Keromytis, Drop-in security for distributed and portable computing elements. *Internet Research: Electronic Networking, Applications and Policy*, 13(2), 2003, pp. 107–115.
8. CERT, Advisory CA-2003-04: MS-SQL server worm, http://www.cert.org/advisories/CA-2003-04.html, Jan. 2003.
9. CERT, Advisory CA-2001-19: "Code red" worm exploiting buffer overflow in IIS Indexing Service DLL, http://www.cert.org/advisories/CA-2001-19.html, July 2001.
10. Y. Bartal, A. Mayer, K. Nissim, and A. Wool, Firmato: A novel firewall management toolkit, in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, May 1999, pp. 17–31.
11. J. Epstein, Architecture and concepts of the ARGuE guard, In *Proceedings of the Fifteenth Annual Computer Security Applications Conference (ACSAC)*, Scotsdale, Dec. 1999.
12. M. Greenwald, S. K. Singhal, J. R. Stone, and D. R. Cheriton, Designing an academic firewall. Policy, practice and experience with SURF, in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, Feb. 1996, San Diego, pp. 79–91.
13. J. D. Guttman, Filtering postures: Local enforcement for global policies, in *Proceedings of the IEEE Security and Privacy Conference*, May 1997, Oatland, CA, pp. 120–129.
14. S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith, Implementing a distributed firewall, in *Proceedings of Computer and Communications Security (CCS) 2000*, Nov. 2000, Athens, pp. 190–199.
15. W. LeFebvre, Restricting network access to system daemons under SunOS, in *Proceedings of the Third USENIX UNIX Security Symposium*, 1992, Baltimore, pp. 93–103.
16. B. McKenney, D. Woycke, and W. Lazear, A network of firewalls: An implementation example, in *Proceedings of the Eleventh Anual Computer Security Applications Conference (ACSAC)*, Dec. 1995, New Orleans, pp. 3–13.
17. J. Mogul, R. Rashid, and M. Accetta, The packet filter: An efficient mechanism for user-level network code, in *Proceedings of the Eleventh ACM Symposium on Operating Systems Principles*, Nov. 1987, Austin, TX, pp. 39–51.
18. J. C. Mogul, Simple and flexible datagram access controls for UNIX-based gateways, in *Proceedings of the USENIX Summer 1989 Conference*, 1989, Baltimore, pp. 203–221.
19. A. Molitor, An architecture for advanced packet filtering, in *Proceedings of the Fifth USENIX UNIX Security Symposium*, pp. 117–126, Saltcate City UT, June 1995.
20. D. Nessett and P. Humenn, The multilayer firewall, in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, Mar. 1998, San Diego, pp. 13–27.
21. W. Venema, TCP WRAPPER: Network monitoring, access control and booby traps, in *Proceedings of the Third USENIX UNIX Security Symposium*, 1992, Baltimore, pp. 85–92.