Chapter

1

Introduction to Firewalls

This chapter provides a brief overview of firewalls—what they can and cannot do. It is not meant to comprehensively cover the topic of firewalls or network security in general. These topics are better covered by more general texts. In this chapter, you will explore some of the technologies used in firewalls, investigate which technologies are used by FireWall-1, and establish why FireWall-1 is the right firewall for you. Examples of how a given technology handles a specific service are also provided.

By the end of this chapter, you should be able to:

- Understand what a firewall is and is not capable of
- Understand what technologies firewalls typically employ
- Discuss the pros and cons of different firewall technologies
- Understand why FireWall-1 is the right firewall for the job

What Is a Firewall?

A firewall is a device that allows multiple networks to communicate with one another according to a defined security policy. They are used when there is a need for networks of varying levels of trust to communicate with one another. For example, a firewall typically exists between a corporate network and a public network like the Internet. It can also be used inside a private network to limit access to different parts of the network. Wherever there are different levels of trust among the different parts of a network, a firewall can and should be used.

Firewalls are similar to routers in that they connect networks together. Firewall software runs on a host, which is connected to both trusted and untrusted networks. The host operating system is responsible for performing routing functions, which many operating systems are capable of doing. The host operating system should be as secure as possible prior to installing the firewall software.

This not only means knowing how the operating system was installed but also making sure that all of the security patches are applied and that unnecessary services and features are disabled or removed. More details about these security issues are provided in Chapter 3.

Firewalls are different from routers in that they are able to provide security mechanisms for permitting and denying traffic, such as authentication, encryption, content security, and address translation. Although many routers provide similar capabilities (such as high-end devices from Cisco), their primary function is to route packets between networks. Security was not part of their initial design but rather an afterthought. A firewall's primary function is to enforce a security policy, and it is designed with this in mind.

What a Firewall Cannot Do

It is important to realize that a firewall is a tool for enforcing a security policy. If all access between trusted and untrusted networks is not mediated by the firewall, or the firewall is enforcing an ineffective policy, the firewall is not going to provide any protection for your network. However, even a properly designed network with a properly configured firewall cannot protect you from the following dangers.

- *Malicious use of authorized services:* A firewall cannot, for instance, prevent someone from using an authenticated Telnet session to compromise your internal machines or from tunneling an unauthorized protocol through another, authorized protocol.
- *Users not going through the firewall:* A firewall can only restrict connections that go through it. It cannot protect you from people who can go around the firewall, for example, through a dial-up server behind the firewall. It also cannot prevent an internal intruder from hacking an internal system. To detect and thwart these kinds of threats, you may need a properly configured intrusion detection/prevention system.
- *Social engineering:* If intruders can somehow obtain passwords they are not authorized to have or otherwise compromise authentication mechanisms through social engineering mechanisms, the firewall won't stop them. For example, a hacker could call your users pretending to be a system administrator and ask them for their passwords to "fix some problem."
- *Flaws in the host operating system:* A firewall is only as secure as the operating system on which it is installed. There are many flaws present in operating systems that a firewall cannot protect against. This is why it is important to properly secure the operating system and apply the necessary security patches before you install the firewall and on a periodic basis

thereafter. It also explains why "appliance" firewalls such as those provided by Nokia and NetScreen, which contain a purpose-built, hardened operating system, are becoming more popular.

- *All threats that may occur:* Firewall designers often react to problems discovered by hackers, who are usually at least one step ahead of the firewall manufacturers.

An Overview of Firewall Security Technologies

Many companies engage in marketing hype to try to prove that their technology is better. Despite the hype, all firewall security technology can be broken down into three basic types: packet filtering (stateful or otherwise), application layer gateways, and Stateful Inspection.

Packet Filters

Packet filters screen all network traffic at the network and transport layer of the TCP/IP packet. This means they look at source and destination IP addresses, protocol number, and, in the case of TCP and UDP, source and destination port numbers. Packet filtering is built into routers as well as some UNIX kernels. Usually, when site administrators start thinking about network security, they start with packet filtering because it is inexpensive. Most routers on the market today, even consumer-grade models, support some form of packet filtering. Because routers are needed to connect different networks together (especially when connecting to the Internet), the additional cost for using this technology is minimal. Packet filtering requires very little extra memory and processing power, so even a low-end router can handle a fairly moderate load. Packet filtering is also fairly transparent to legitimate users.

Traditional packet filtering is static, that is, the only criteria for allowing packets are whether or not the IP addresses or port numbers match those specified in the packet filter configuration. Many packet filters today implement some concept of the "state" of a connection, using a table and additional information in the TCP headers to track previously allowed packets within a connection. This makes it much easier to allow only, for instance, outbound connections from a trusted network to an untrusted network without inadvertently allowing unrelated packets from the untrusted network to the trusted network.

The biggest downside to packet filters is that they are difficult to maintain. Although this point is certainly arguable, even an expert can have trouble configuring a moderately complex set of access lists or Linux `ipchains` rules. Many consumer-grade routers that have packet filtering do not have an adequate interface or are very limited in what they can filter.

Packet filters also do not screen above the network and transport layers. This means they cannot do things like:

- Provide content security (e.g., virus scanning or filtering based on specific sites and Web pages accessed)
- Authenticate services (i.e., make sure only authorized users use a service)
- Dynamically open and close ports for applications as they require them (necessary for applications like RealAudio, FTP, and H.323 applications)
- Validate a particular port that is used only for a specific service (e.g., making sure that only valid HTTP traffic traverses port 80)

Application Layer Gateways

Application layer gateways, also known as *proxies* or *application proxies,* take requests from clients and make them connect to servers on the client's behalf. In some cases, the client explicitly connects to the proxy server. In other cases, the proxy intercepts the connection with help from the underlying operating system or network architecture. Because an application proxy is usually specific to the network service, it can be fully aware of the session. This means the proxy can do content screening, provide authentication, and ensure that only the particular service is used (e.g., an HTTP proxy can make sure that only HTTP traffic is allowed through), or it can provide other application-specific services such as caching. It also provides a well-formed connection to servers on the other side of the firewall because it opens up connections on behalf of the clients.

However, this extra capability comes at a price. Application proxies require memory and CPU cycles just like any other application. Generally speaking, application proxies use more memory and CPU cycles than packet filtering, although how much they use depends on the specific circumstances. If you want to use application proxies to provide services to the Internet, each application you want to run through your firewall must have a proxy written for it, or the application must be compatible with a "generic" proxy that will work with simple TCP or UDP connections. Because many applications are not being developed to work with an application proxy, some applications simply cannot be proxied. The client/server model is somewhat broken by application proxies because the application proxy will always originate the connection from the server's point of view.[1] In large environments, the poor throughput of application proxies is another drawback.

1. Some would argue that this is actually not a problem. Whether or not this is a problem depends on the specific application and what you are trying to track down.

Another important drawback of a proxy, particularly for internal use, is that it becomes very difficult to track who is going where for how long because the proxy often masks the original source or destination of the traffic. You might be able to track this on the firewall, but from any other vantage point on the network, how do you know?

Stateful Inspection

Stateful Inspection combines the best features of stateful packet filtering and application layer gateways. Check Point's Stateful Inspection engine rests between the data link and network layers (e.g., between the network interface card and the TCP/IP driver). TCP/IP packets from the network layer and higher are scanned according to your security policy and will be either allowed through or stopped. The TCP/IP stack will not see dropped or rejected packets, which can provide an extra layer of protection. Stateful Inspection can look at the entire packet and make security policy decisions based on the contents and the *context* of the packet, using a state table to store connection state and using knowledge of how specific protocols are supposed to operate. In the case of FTP, FireWall-1 can dynamically open ports between two hosts so that the communication will succeed and then close the ports when the connection is done. Stateful Inspection is what gives FireWall-1 "Application Intelligence" (e.g., NG with Application Intelligence, or NG AI).

Stateful Inspection requires slightly more memory and CPU cycles than packet filtering because it has to do more, but it takes substantially less memory and CPU usage than does an application proxy. Stateful Inspection is best when the engine is made aware of how a protocol functions, although Check Point does not make use of Stateful Inspection for every protocol. Because Stateful Inspection does track connection state regardless of the service, it is better than a packet filter, but you are limited to opening specific ports and allowing the traffic through without further checking.

Technology Comparison: Passive FTP

It is useful to compare how the different technologies handle complex connection types. One such connection type is Passive FTP, which is used by Web browsers when they initiate an FTP connection. Passive FTP requires:

1. A TCP connection from a client to port 21 on the FTP server.
2. A TCP connection from a client to some random high port on the FTP server for data communication. The ports used for this communication are communicated to the client when it requests passive mode via the PASV command.

For this comparison, assume that the FTP server is behind your firewall and that you need to allow people on the Internet to FTP to this machine.

Packet Filters

Packet filtering can handle standard FTP quite nicely because it uses fixed TCP ports (20 and 21). However, in order to allow Passive FTP, the packet filter has to open all TCP ports above 1024 to allow Passive FTP to work with the FTP server. This is a gaping hole that can be used by programs other than FTP to compromise your systems.

Application Proxies

An application proxy is aware of the FTP connection and opens all the necessary ports and connections to complete the FTP connection. However, each TCP or UDP connection through an application proxy requires twice the normal number of connections on the proxy server (one for each side of the connection). A normal Passive FTP connection requires two open connections on a client machine. On the application layer gateway, this translates to four open TCP connections.

Most operating systems have a limit to the number of simultaneous connections they can handle. If enough connections are going through the machine at the same time, this limit will be reached, and no further connections will be allowed through. In high-performance, high-capacity networks, using a proxy for FTP connections is simply asking for trouble.

Stateful Inspection

Stateful Inspection understands connection context. When the PASV command is sent from the client to the server, Stateful Inspection reads the server's response and opens the ports necessary to complete the connection. It also restricts the IP addresses that can use these ports to the client and server. The connection then goes through the firewall normally. Because Stateful Inspection allows the native operating system to route, no connections are established on the firewall itself. Once the connection is terminated, the ports opened by the PASV command are closed.

Technology Comparison: Traceroute

Traceroute is used to show the particular path a connection will take through the various routers and gateways within the network and gives you a basic idea of the latency between any two hosts on a network. It is a common troubleshooting tool used by network administrators. There are two varieties of traceroute:

UDP and ICMP. UDP traceroute is used by almost every UNIX implementation. ICMP traceroute is typically used by Microsoft operating systems, though some UNIX implementations also allow you to perform an ICMP traceroute. How traceroute functions can be used to show the strengths of Stateful Inspection and the weaknesses of packet filters and application proxies.

UDP traceroute involves sending out packets to high-numbered ports above 31000—the actual ports used will vary based on the implementation. ICMP traceroute uses ICMP Echo Requests instead. In both cases, the client generates a number of packets (usually three) over a period of time (usually one second) to the server using a time to live (TTL) value of 1. Each subsequent set of packets will have an increasingly higher TTL value, which allows the packets to get closer and closer to the server.

During a traceroute session, any of the following can occur.

- The server responds with an ICMP Echo Reply message or an ICMP Port Unreachable packet (i.e., the traceroute has finally reached the server).
- An intermediate router or gateway gets a packet with a TTL value of 1; it decrements the TTL to 0. Because a router or gateway cannot route a packet with a TTL of 0, it sends back an ICMP Time Exceeded message.
- An intermediate router or gateway determines it has no route to the server and sends back an ICMP Destination Unreachable message.
- An intermediate router or gateway fails to respond either because it is configured to not respond to or pass traceroute traffic or because it is down.
- The client decides it has sent too many sets of traceroute packets (the default is 30) and stops.

For any firewall solution to securely allow traceroute through,[2] it must take all of these situations into account. Let's explore how each of the firewall technologies can address passing traceroute.

Packet Filters

With packet filtering, you would have to allow the following types of traffic to pass through your packet filter:

- All UDP ports above 31000
- ICMP Echo Request

2. Most firewall administrators do not want to allow traceroute into their network from the outside but do wish to allow internal hosts to initiate it outbound and then allow only appropriate reply traffic back in.

Conversely, you would also have to allow the following types of packets to enter your network from any host:

- ICMP Echo Reply
- ICMP Time Exceeded
- ICMP Destination Unreachable
- ICMP Port Unreachable

Although these rules would allow legitimate traceroute traffic, they can also allow network access by packets that were not in response to a valid traceroute request. In the past, these kinds of unsolicited packets were used in denial-of-service (DoS) attacks. It is important that you allow in only those packets that are in response to a traceroute or ping query. Packet filtering alone is not an adequate tool to allow traceroute to function yet protect you from possible DoS attacks. It is important to note that the UDP ports allowed could also be used for something other than traceroute.

Application Proxies

UDP can be proxied to some degree, but due to its nature, ICMP cannot be proxied, though some versions of SOCKS can proxy ICMP using special SOCKS-aware ICMP programs. In a relatively small, controlled, homogeneous environment, this may be feasible. In a large, heterogeneous environment protected by application proxies, it may not be possible to allow all clients to traceroute through the firewall.

Stateful Inspection

With Stateful Inspection, you can watch for either a UDP packet with a low TTL value or an ICMP Echo Request packet coming from a particular client. Once this happens, you can temporarily permit the necessary ICMP packets to return to the client initiating the outgoing traceroute request. After you have received the appropriate response (i.e., an ICMP Echo Reply, Port Unreachable, or Destination Unreachable message) and/or after a specific period of time (e.g., 60 seconds), you can stop allowing the necessary ICMP packets to the client.

FireWall-1 statefully inspects ICMP.

What Kind of Firewall Is FireWall-1?

Check Point advertises FireWall-1 as primarily a Stateful Inspection firewall. Although this is certainly FireWall-1's biggest strength, FireWall-1 uses both Stateful Inspection and application proxies. Application proxies are used when

content security or user authentication is necessary for HTTP, Telnet, rlogin, FTP, and SMTP. Stateful Inspection is used for all other security functions. To be fair, most commercial and even homegrown firewalls employ some combination of these two technologies because none of the technologies can provide all the necessary functionality.

FireWall-1 also offers some other interesting capabilities, many of which are covered in future chapters:

- Site-to-site VPNs
- Client-to-site VPNs
- Content filtering (with the help of third-party products)
- Address translation
- Authentication (integrated with third-party authentication servers)
- Enterprise-wide policy management
- High availability (with the help of third-party products)
- INSPECT, a language with which you can modify Check Point's Stateful Inspection engine

Do You Really Need FireWall-1?

Whether or not you really need FireWall-1 might seem like a strange question to ask in a book about FireWall-1. One of the important points I make in this book is that FireWall-1 is simply a tool used to enforce a security policy. In some cases, using this tool may be overkill. In other cases, this tool is just one of many that are used.

Let's look at a one- or two-person site. In this case, whether or not to use a firewall depends on what the network connection is and what needs to be protected. If the connection is an analog dial-up connection to the Internet that does not stay up a majority of the time, a firewall may not be entirely necessary. If the connection is something more permanent, like a leased line, Digital Subscriber Line (DSL), or cable modem, or if what goes on at this site is highly sensitive or valuable, a firewall may be necessary. If the people who occupy this site are technically savvy, perhaps they will set up their external router with an access list, set up a multihomed host using a BSD or Linux-based operating system, use one of the many consumer-grade firewall devices on the market, or install personal firewall software on the computers. Depending on what the site's needs are, these solutions may be sufficient.

Now let's look at a slightly larger site, say, one that employs 25 to 50 people. This type of site is likely to have some sort of permanent Internet connection.

It may even have an externally accessible server or two like a mail server and a Web server. Again, as mentioned previously, this type of site could probably get away with setting up a multihomed host using a BSD or Linux-based operating system running their built-in filtering mechanisms, or an access list on a router. Perhaps the site also needs to allow one or two people access to the internal network from home. At this point, a few "holes" would be added to the firewall. At a later time, a few other people might want to use some sort of specialized application through the firewall and a few more holes would get added. Pretty soon, the firewall starts to look like Swiss cheese.

Now let's talk about a large corporate site with thousands of people. A site like this could use a firewall or two. One obvious place to put a firewall would be at the external connection to the world, but firewalls could also be used internally to protect certain sensitive departments like human resources, research and development, or accounting. And perhaps this corporate site is also responsible for some smaller remote offices. These remote offices likely need secure access into the internal network at the corporate site. Also, the corporate site might like to be able to manage the security policy for the remote sites. And, of course, there are those who want to work from home or who need secure access to the corporate network from the Internet.

People tend to think of security needs in terms of the size of the network involved. The preceding examples are typical of what I have experienced in the real world. What type of firewall you require, if any at all, really comes down to your specific needs or the needs of an organization. A one- or two-person site might be developing source code that could potentially be worth millions of dollars; thus network security becomes important. Another example might be a university network with thousands of students, where an open environment is far more important than a secure environment—although you can bet that certain parts of the network, like admissions and finance, require very tight security. The main question you have to ask when considering a firewall is, "What is at stake if an unauthorized person gains access to my network?"

FireWall-1 is an appropriate solution for networks of all shapes and sizes. This is because FireWall-1 is one of the few firewalls that can grow with your needs. In a network with few needs, FireWall-1 can start out as a simple Internet firewall. As your needs change, you can easily add firewalls and still be able to easily keep track of and manage your corporate-wide security policy. As your network grows, you can readily upgrade or change the platform on which FireWall-1 is installed and add functionality, such as a VPN, quite easily. Because FireWall-1 works the same on all supported platforms, you will not have to spend a significant amount of time reconfiguring or relearning the product. Adding new functionality is usually as simple as adding a new license string and

modifying your configuration to support the new feature. With the added functionality of INSPECT, you can program FireWall-1 to securely support just about any service.

With the help of this book, you will be able to effectively use FireWall-1 in just about any network environment in which you work.

More Information

Many other network security topics could have been covered in this chapter, and even the topics covered could have been covered in greater depth. However, the focus of this book is not on general security topics but rather on FireWall-1. Many of the general topics are covered in depth in other books by other authors. Appendix G includes a list of Web sites with more information on interesting software. Appendix H includes a list of recommended books.