An Introduction to Internet Security and Firewall Policies

William Hugh Murray

Payoff
This article is an introduction to security on the internet. It describes the characteristics, applications, and protocols of the network. It also describes and explains the peculiar vulnerabilities that arise from these characteristics and the attacks that exploit them. This article offers strategies, tactics, and mechanisms for protecting the traffic on the network. It places special emphasis on firewalls and encryption and strategies for using them.

Introduction
Any attempt to describe anything as dynamic, not to say unstable, as the Internet, is likely to make one look foolish. Describing the Internet can be liken to five blind men trying to describe an elephant. However, the elephant remains an elephant, it does not change during the examination and discussion. On the other hand, descriptions of the Internet that are only three years old are already so out of date as to be inaccurate if not dangerously misleading.

The Internet is already the most complex artifact in history. It may turn out to be important, or it may not. On the chance that it is or will be important, it makes sense to try to understand it, no matter how difficult and uncertain an explanation is likely to be.

The Characteristics of the Internet
The Internet can be defined and described, in part, in terms of its characteristics. Although it is possible for a network to have some of these characteristics without having them all, they are related in subtle ways.

Public and Open
Perhaps one of the most important characteristics of the Internet, at least from a security point of view, is that it is essentially public and open. It is public in the sense that, like the phone system, anyone can use it. One may have to go to a pay phone, a kiosk, or the public library, but anyone can use it. Libraries have been known to hand out user IDs with the same frequency as library cards. No requirements exist to be able to use the Internet, i.e., anyone can use it. In addition, as in broadcast TV, radio, or magazine advertising, most of the traffic is public. Its value increases with the number of people who see it. Although it has not always been so, most of the servers and services available on the Internet do not know or care who their users are. No user identification or authentication is required. The servers may count the accesses and they might like to know the demographics of those who visit, but otherwise, the greater number of visits, the more successful the site is considered.

Similar to it being public, the Internet is open. Like the postal system and for the price of a postage stamp, anyone can send a message. For the price of an accommodation address, anyone can receive a message. Although there may be an agreement to pay, no other permission is required and, as a rule, payment in advance is not required. The Internet is also open in the sense that with a minimum of notice to or cooperation of others a connection can be made. A node at the edge of a network can be added easily and unilaterally, creating a new connection between networks. Therefore, it is difficult, nearly impossible, to know what the network looks like.
Although only a small percentage of the traffic on the Internet is sensitive to disclosure and most applications and services are free, almost all traffic is sensitive to contamination and most services are sensitive to interference. Moreover, although many who offer public information on the Internet want many people to see it, they want it to get through in tact; they do not want it modified, they do not want it broken, and they do not want to be responsible for what they did not say. The public and open nature of the Internet makes this more difficult to achieve. It also makes it more difficult to achieve confidentiality and accountability for that traffic and those applications that require them.

**Inclusive Network of Networks**

By definition, an internetwork is a network that connects networks. Therefore, the Internet is a network of networks. It is one collection of all networks, and the economic advantage of a connection is so great as to be irresistible. Moreover, although isolated networks may exist in the short term, in the long term, the internetwork will be one. Isolated networks that persist will be sparse, small, and temporary as not to be significant.

**Mesh Topology**

The Internet has a mesh topology, which means that, except at the edges, most nodes are connected to two or more other nodes. In addition, there are multiple paths between any two points on the network, because the topology maximizes the potential that a message will get through and maximizes the total message carrying potential (i.e., bandwidth) of the network. On the other hand, at least by default, users do not know what path their traffic will follow or what nodes and links their messages will traverse.

**Flat**

Ideally, the Internet is flat, as opposed to hierarchical. Information flows directly from the origin to the destination rather than in, to a central switching point, and then back out to the destination. Therefore, the cost to send a message between any two points on the network is the same as between any other two points. The time required for a message to move between any two points is roughly the same as for any other two points chosen at random. Finally, the bandwidth between any two points is roughly the same as for any other two points.

As expected, messages flow more quickly between nodes that are close together. However, it is possible for a part of a message to circle the globe, even when addressed to a nearby node. So, at least on average, across all randomly chosen pairs of nodes, the Internet is flat.

**Broadcast**

A node that desires to send a message to another node broadcasts that message to the remainder of the network. Depending on the routing algorithm used, the originating node may prefer nodes that it thinks are in the direction of the destination. However, it is possible for a message to traverse the globe even when addressed to a nearby node. Other nodes that receive the message look at the destination address in the message and forward it in the general direction of that destination. This is similar to a point-to-point network in which the path between two points is determined in advance and dedicated, at least for the instant, to carrying that message. Although every packet does not pass every node and it is possible for users to influence the path that their traffic follows, few users have the necessary special knowledge to take advantage of this capability. They do not know how to exercise the control or to distinguish one path from another. Such control, if used, would
limit the paths and bandwidth available to the traffic and be achieved at the cost of a reduction in the chances that the traffic would get through quickly.

**Different Types of Internet Connections**

Three kinds of connections are available on the Internet.

**Packet-Switched.**

Related to the idea of broadcast is that of packet-switched. A message is broken into packets, each packet is labelled as to its origin and destination and then is broadcast onto the network. Other nodes forward the packet in the general direction of the destination. It is possible that adjacent packets in a message will follow different paths to the destination. This is the opposite of circuit-switched networks, such as the voice network, in which a circuit or path is determined in advance and all parts of the message follow the same path. In a packet-switched network, an intervening node may see only a part of a message. On the other hand, it increases the number of nodes that may see a part of it.

**Peer-Connected.**

Nodes on the Internet are “peer connected.” No node dominates or controls another. Thus, by default, all nodes behave as if they trust all other nodes as themselves. The implication is that the level of trust is equal to that of the least trusted node.

**Any-to-Any Connection.**

Like the postal system, and except as otherwise restricted, any device connected to the Internet can send a message to any other device. There is no requirement for an answer but, at a minimum, the destination device must recognize the message and make a decision about it. For example, at MIT the softdrink vending machines are connected to the Internet. If the addresses are known, they may be queried from anywhere in the world.

**Increasing Interoperability**

If connectivity is the ability to send a message to any node, interoperability is the ability to get a meaningful answer back. Already, the Internet is better at answering questions than most individuals are at asking questions. The Internet can provide a report of freeway traffic in Los Angeles, hotel availability in London, or the schedule of every opera house in the world for the next two years. It can also locate all the bed and breakfast lodgings in most places in the world, and get an index to the treasures of the Vatican Library or of the British Museum. Individuals can locate and download graphics, moving images, and general and specialized software. A query on “Mona Lisa” returns references to both 1000 different prints of Da Vinci’s La Gioconda and a sound clip of the Nat King Cole song. If the necessary software is unavailable to interoperate with another system at a particular layer, software can be downloaded at another.

As protocols and interfaces become more standard, they become more useful. As the use of a standard increases, so does the propensity to comply with it. The less standard an interface, the more it must include information about its intended or productive use.

**No Central Authority**

Although there are authorities such as the Internet Activities Board (IAB) and the Internet Engineering Task Force (IETF), which make architectural and design decisions for the Internet, no one is obliged to follow them. The individual networks are independently
owned and operated. There is no central authority that is responsible for the operation of the entire network. Because the network is global, it is not even subject to the authority of any single nation state.

Internet Protocols

The Internet can also be defined and described in terms of the communication protocols that it employs. One, somewhat pure, definition is that the Internet is that collection of interconnected networks that employ the Transmission Control Protocol and Internet protocol and transmission control protocol/Internet protocol (TCP/IP) suite of protocols. A more practical definition is that the Internet is that set plus those networks connected to it by appropriate gateways. (For purposes of this definition, a gateway is a node that translates traffic from one protocol to another.)

The Internet Protocol

The fundamental protocol of the Internet is IP, the Internet protocol. IP is the network layer protocol for the TCP/IP Protocol Suite. It is fundamental in the sense that all other protocols are built on it. It is connectionless, best-effort, packet-switched, and unchecked. “Best effort” means that the network will do its best to deliver the packet, but there are no guarantees. “Unchecked” means that there is no redundancy in the protocol to enable either the sender or the receiver to know whether the packet was received correctly. There is no acknowledgement of the receipt of the message. The receiver cannot be sure that the message comes from where the origin address of the packet says that it comes from.

IP is to the Internet as the post card is to the postal system, limited in capacity, function, and intent. However, just as a message of any length can be sent by using multiple post cards, or by using one post card to acknowledge or to check on another, IP packets can be composed in such a way as to compensate for all of these limitations. These compositions make up the higher-level protocols.

The Transmission Control Protocol

The transmission control protocol (TCP), is the standard Internet protocol (IP) for the transfer layer. It defines how IP packets are sent back and forth between a sender and a receiver to provide many of the things that IP does not. However, even TCP does not provide security nor the reliability of origin and destination. Both the sender and the receiver know that they are talking to someone that is orderly and well behaved, but they do not know for sure that it is their intended party, and they do not know if any one is listening in.

The Oldest and Most Widely Used Protocols

The following are among the oldest and most widely used protocols on the Internet:

- **Telnet.** This was originally intended for connecting host-dependent terminals to remote systems or applications. Today, it is used by terminal emulator programs on workstations.
- **File Transfer Protocol.** FTP is used to move files from one system to another.
- **Simple Mail Transfer Protocol.** SMTP is used for E-mail.

The applications of these protocols are discussed in subsequent sections.
Other Common Protocols

In addition to those protocols previously discussed are the following:

· **Serial Line Internet Protocol.** The Serial Line Internet Protocol is used to exchange IP traffic with a device, usually a workstation, that is running the proper protocols but without a separate address. It is used to connect workstations to hosts or to Internet service providers through the dial-switched network. It is analogous to an extension cord or a remote.

· **Point-to-Point Protocol.** The Point-to-Point Protocol is similar to Serial Line Internet Protocol, but is associated with leased lines. It is usually used to connect a single system to a boundary or “edge” node.

· **Network Time Protocol.** The network time protocol (NTP) is used to set and synchronize the system clocks of Internet nodes. It is able to synchronize all systems in a network to within milliseconds of each other, i.e., to within the accuracy and precision of the system clocks themselves.

· **Secure Protocols.** Recently, secure versions of these protocols have been specified, and reference implementations of these protocols are available for Unix systems. Additional implementations should be available in 1996.

Internet Applications

Recall the analogy that describing the Internet can be liken to five blind men trying to describe an elephant. For most of the blind men, the Internet elephant looks like its applications. The Internet is open as to its applications. No real limit to the number of applications exists, and new ones are added every day. However, some applications are sufficiently significant that a description of those applications describes how the net looks to most users.

**E-mail**

The most widely used application on the Internet is E-mail. Recent statistics suggest between 50 and 100 million users, and a 1 billion users are estimated as early as 2000. E-mail rivals television, copiers, and facsimile machines in its rate of growth. Moreover, as was the case with copiers and facsimiles, it is becoming difficult to remember how business was conducted before E-mail.

Internet E-mail uses the simple mail transfer protocol (SMTP), and the multipurpose Internet mail exchange (MIME) protocol. MIME runs on top of SMTP to permit the exchange of files, programs, sounds, images, and moving images. E-mail is the most interconnected and interoperable application. Even those networks that have resisted connection to the Internet at other levels are connected at the E-mail layer.

In addition, E-mail is the most ubiquitous application in the Internet; it interoperates with many of the others. Several servers are on the Internet that accept mail messages, convert them into requests for other services, convert the answers to those mail messages, and send them back to the requestor. Thus, a user that has access to E-mail functionality, has access to all of the information on the network (i.e., Internet).

**Logging on to a Remote System**

One of the earliest and most obvious of Internet applications was to create a session between a terminal on one system and an application on a remote system. This kind of
application used a client process on the origin system, the Telnet client. It IS-IS initiated by entering the command, `telnet`, on the originating system. The parameters of the command specify the target system and any nondefault characteristics of the connection request. The request is responded to by a the telnet server, a started process (a daemon in Unix parlance) on the target system. The protocol is also called telnet. The user on the origin system sees a prompt from the answering server process, for example, the operating system or an application, on the target system. The user is usually expected to logon, that is, send a user identifier (i.e., user ID) and authenticating data (i.e., a password) to the target system. However, for the target system, the user identifier and password are optional.

**File Transfer**

The File transfer protocol (FTP) is used to exchange file system objects between systems. It is symmetric, and works in either direction. Either system may initiate a transfer in either direction. The File Transfer Protocol process (daemon in Unix parlance) must have access to the file system. That is, in systems with closed file systems, the process or the user on whose behalf it is operating must possess the necessary access rights (e.g., read, write, or create) to the file object or directory on which it wants to operate.

A convention called, “anonymous FTP,” permits the protocol to be used for public applications. The user can logon to the system with a user ID of anonymous, which requires no password. By convention, users are requested to put their origin system and user ID in the password field. However, the value in this field is not checked or validated in any way; a blank will work as well as the truth.

**Vulnerabilities on the Internet**

The vulnerabilities on the Internet are closely related to its characteristics, its protocols, its uses, and its history. In addition, because the Internet is a broadcast network, messages are vulnerable to disclosure, replay, and interference.

The large number of components on the Internet makes it vulnerable to flaws in the implementation of those components. Because there may be many instances of a flaw, elimination of them is extremely difficult. A recent example of such a flaw was an instance of incomplete parameter checking in the Unix system logging routine, `syslog`. This error permitted a very long log entry to exceed the space provided for it, overlay program space, and get itself executed.

Many components in systems peer-connected to the Internet contain “escape” mechanisms. These are invoked by an otherwise unlikely character sequence to cause what follows this escape sequence to be handled, not by the component itself, but by the environment in which it runs, often with the privilege of the “escaped from” component. A famous escape mechanism, exploited by the infamous “All Souls” worm, was the debug feature of the sendmail mail handler. This option was invoked by an escape sequence in a message that caused what followed it to be passed through to Unix to be executed as a command. The worm used this feature, among others, to copy and execute itself.

Because nodes are peer-connected and trust each other, compromise of one may result in compromise of many, perhaps all. In a peer-connected network, the level of trust in the network is equal to that of the least trusted node or link.

Many of the vulnerabilities described in the preceding paragraphs are features rather than flaws. In other words, they are desired and valued by some users and managers. Because of their value, their total elimination is unlikely.

Every node on the Internet has a system manager or privileged user. This user is not subject to any controls intended to ensure that users and their systems are orderly and well-behaved. In single user systems, the only user is a peer of the privileged user in the multi-user system. That user is assumed to have the same motivation, training, and supervision
as the manager of a multi-user system. The vast number of such users ensures that at least some of them will be disorderly and unreliable. Because they are all peers and because the systems are peer connected, it makes little difference which of them are trustworthy.

The Internet is so large and complex that no one, not the designers, not the implementers, and not the users, fully apprehends it, much less comprehends it. Everyone are the blind men. Nonetheless, its immense scope and size make it unlikely that it will ever be perfect. Attackers look on it as a “target rich” environment. Although most nodes on the network are implemented, configured, and operated so as to resist attack, the great number of them ensures that there will always be some that are vulnerable to attack.

Finally, two of the vulnerabilities on the Internet, insecure links and insecure nodes, are fundamental. In other words, they are inherent to the Internet, nature, use, intent, or at least its history. Contrary to popular belief, they are not the result of errors, flaws, or failures on the part of the designers, implementers, or operators of the network. Rather, these insecure links and nodes are the result of attempts to have the greatest chance of getting a message from point A to point B in the least amount of time. They are never going to go away; it is not simply a matter of time. Indeed, at least for the next five years, they are likely to get worse. That is, vulnerabilities will be increase faster than the ability to fix them. Moreover, the number of insecure links and nodes in the network are both growing at a much faster rate than the number of secure ones. This vulnerability is certain and extremely resistant to change.

**Attacks on the Internet**

The conditions for a successful attack include necessary access, special knowledge, work, and time. Because of its nature, all of these things are somewhat more available on the Internet than on other networks. Because the Internet is open, almost anyone can gain access. Most of the special knowledge in the world is recorded, encapsulated, and available on the Internet, mostly for the taking; although, every now and then permission is required. Even much of the necessary work to launch a successful attack has been encapsulated in computer programs. Thus, they can be perpetrated by those who lack skill and special knowledge and who are not prepared to do the work themselves.

**Eavesdropping**

As packets move through the net, they can be observed by privileged users of the nodes or by using special equipment to listen in on the links. These attacks are easily automated.

**Packet and Password Grabbers**

A packet grabber is an automated eavesdropping attack, a program that copies packets as they move through an intermediate node (i.e., a node between the origin and destination). A password grabber is a special case of a packet grabber that identifies and stores for later use user IDs and passwords as they pass through an intermediate node. Because, at least as a general rule, unprivileged processes cannot look at traffic in transit, password grabbers must be installed by privileged users. However, recent experience suggests that they are often placed in penetrated systems. Writing password grabbers requires special knowledge and work. However, now, so many copies of those programs exist that the attack can be used even by those without the knowledge and not prepared to do the work. The Internet has so may password grabbers that passwords in the clear are not sufficiently reliable for commercial or other sensitive applications, and the problem moves from the category of an attack to that of a pervasive problem.
**Address Spoofing**

The origin address on the IP packet is not reliable. The sending system can set this address to any value that it wishes. Nonetheless, by convention and for convenience, many systems rely on this address to determine where a packet came from and to decide how to treat it. Packets carrying the origin address of recognized systems may be treated as though they had originated on a trusted system. Again, with sufficient work and knowledge, it is possible to write a program to exploit this trust. Toolkits for building this kind of attack have been written and distributed within the hacker community.

**Trojan Horses**

A Trojan Horse attack is in one in which a hostile entity, for example, armed warriors, is concealed inside a benign or trusted one, for example a gift horse, to get it through a protective barrier or perimeter, in the original case, the walls of the city of Troy. In computer science, it usually refers to a malicious program included in another program or even in data. Although most systems are vulnerable to this kind of attack to some degree or another, and it has always been a concern, until the proliferation of desktop computers and viruses, it was not a problem.

As previously discussed, both node-to-node connectivity and trust and open file systems make the Internet particularly vulnerable. Trojan Horses can and do travel over any of the popular protocols and in any of the popular object types. For example, they can travel in files over File Transfer Protocol, as documents over MIME, or in arbitrary objects called by HTML scripts fetched from World Wide Web (WWW) servers by browsers. Although some browsers and interpreters (e.g., HotJava) are designed to resist such attacks, most are not. Even in situations in which the browser or interpreter is resistant, it is always possible to dupe some users in a large population.

Trojan Horses are easily executed because they have attractive names or descriptions or the names of frequently used programs. They may require a minimum of user cooperation. For example, the PRANK (virus) was implemented as a MS Word macro and could spread in any Word document. Simply asking Word to open an infected document would contaminate that copy of Word and any document that it subsequently opened. If an infected document were attached to an E-mail message, an act as simple as double clicking the icon for the document would be sufficient to execute the macro. Because such a macro can contain and call an arbitrary program, there is no limit to the sophistication of the program or the contamination it can cause.

Trojan Horse attacks are of special concern on the Internet because they compromise trust of end-point nodes, of the net, and of applications on the net.

**Browsing**

Browsing is going through the network to look at available, public, accidentally, and erroneously available data in search of something of value. Specifically, in an attack sense, this search method looks for special data that will reduce the cost of an attack against other nodes. For example, many systems implement or provide directory services. These directory services return the names of enrolled users, i.e., user identifiers. The information returned by these public services is used by the attacker to identify targets and thereby reduce the cost of attack. Attackers also use browsing to identify and download attack programs.

**Exhaustion**

When confronted with good security and when all other attacks fail, an attacker can always fall back on trying all possible combinations of data(e.g., user identifiers and
passwords) until he or she finds one that gets through. Traditional systems resisted such attacks by disconnecting disorderly devices (i.e., devices that failed to successfully logon). Because the Internet is a broadcast network, there is no connection to break. A system must look at every packet addressed to it and make a determination as to what to do with it. It is possible to spread the attack over time or across addresses so as to disguise the attack as errors or noise.

Denial of Service

Denial of service attacks are those that cause failures by overloading or consuming all available resources. On the Internet, this class of attack includes “spamming” or overloading a target with unwanted traffic. Although the target is not damaged in any permanent way, it may be unable to provide critical services to those intended to use it.

Defending Against Attacks on the Internet

A vast number of options exist that the implementers, operators, and users of the net can do to limit these vulnerabilities and the attacks against them. However, in considering them, keep in mind that these vulnerabilities are fundamental to the nature of the Internet. The only way to eliminate all of the risk is to either eliminate the Internet or alter it so fundamentally that it will lose its identity. Clearly, neither of these options are viable. Rather, the defenses should be balanced against the vulnerabilities so as to preserve essential trust. Discussions of some broad categories of defense mechanisms follow in the subsequent section.

Isolation and Compartmentation

Of course, the most obvious defense against network attacks is simply not to attach, to connect, or to participate in a network. Not only is this defense effective, it is also demonstrable to the satisfaction of third parties. However, the value of the security obtained rarely compensates for the lost value of connecting or participating in a network. Moreover, it has often been said that sensitive defense systems are safe because they are not connected to public networks.

Because the value of connecting to a network is high and because the cost of that connection is low, isolation is difficult to maintain. Even a very small system or a single desk-top workstation can form a connection between networks.

Policies

In the presence of known connections, people can provide protection. They can recognize attacks and take timely and appropriate action. However, for this to be effective, it must be planned and pervasive. If management wishes to rely on individuals', in advance, it must tell them what action to take. A policy is an expression of management's intention. It should contain a recapitulation of the user behavior that management relies on. It should also clearly delineate the responsibilities of employees and managers. Finally, it should specifically address the responsibility to report anomalies.

Bastions

Bastions are “projecting” fortifications. They are strong systems that can be seen from the outside (i.e., the public network), but which are designed to resist attack (e.g., by recognizing only a very limited repertoire of application specific commands). Bastions normally hide the generality and flexibility of their operating systems from the network. A
A full-function gateway system that can be seen from the public network is called a bastion host. Such a gateway must be able to protect itself from its traffic. Finally, because most protective mechanisms can be bypassed or circumvented, all applications and services that can be seen from the network should be able to resist their traffic.

Filters

Filters are processes that pass some traffic while rejecting some other traffic. The intent is to pass safe traffic and to resist attack traffic. Filters may operate on headers or content. Many filters operate on the basis of the origin address in the header. They pass traffic that appears to have originated on recognized or trusted systems. They may also operate on a combination of origin, protocol, and destination. For example, they may pass mail traffic from unknown origins to the mail port on the post office machine and reject outside traffic addressed to the Telnet port on the same machine. Filters are important. For further information see the subsequent section.

Wrappers

Wrappers are proxy programs or processes. They can be viewed as traffic filtering programs. They are designed to protect the target from unintended traffic, known attacks, or to compensate for known weaknesses. They often assume the name of the process that they are intended to protect (i.e., common functions or known targets). For example, suppose that a privileged program is known to have a flaw or an escape mechanism that can be exploited by a packet or a message. A wrapper can be given the name of that program, placed ahead of it in the search order, and used to protect against messages of the dangerous form. After eliminating all messages of the dangerous form, the remainder are passed to the “wrapped” program as normal.

Using a wrapper is a preferable alternative and it presents a lower risk to cure a vulnerability than patching or replacing the vulnerable program. They have been employed to great advantage in Unix systems in which it is often easier to use the wrapper than to find out whether the particular version of Unix or one of its subsystems that is being used has a particular problem. The most famous wrappers are a collection known as COPS. These are used to protect Unix systems from a set of known attacks and vulnerabilities.

Filters: the Most Popular Defense

Filters are the most popular defense to ward off network attacks. The intent is to pass normal traffic while rejecting all attack traffic. Of course, the difficulty is in being able to recognize the difference between the two. Filters are normally based on the origin, the destination, and the kind of traffic. Traffic is permitted to flow from trusted or known sources to safe or intended destinations. Of course, most destinations will ignore traffic that is not addressed to them but will certainly listen to attack traffic that is addressed to them. Filtering on destination address can protect the system from seeing attack traffic at the expense of protecting it from all traffic.

Filters Implemented by Using Routers

In part, because networks are usually connected to each other through routers, routers are a favorite place to filter traffic. The same logic that is used by the router to decide where to send traffic can be used to reject traffic (i.e., to decide to send it to the “bit bucket.”) For example, only those packets that appear to have originated on systems whose addresses are recognized (i.e., on a list of known systems) may be accepted.
Packets by Address: IP Address and Port

A filter must have criteria by which to decide which traffic to pass and which to reject. The criteria must appear in the packet. The most frequently used criteria are the IP origin and destination addresses. Typically, this is expressed as an address pair. In other words, traffic appearing to originate at $A$ and addressed to $B$ may pass this router. Although it could say all traffic originating at $A$ may pass or all traffic intended for $B$ may pass, this is significantly less rigorous or secure.

The origin and destination are usually expressed as IP addresses and may be further qualified by port. That is traffic originating on the mail port of $A$ may pass to the mail port on $B$, but to no other port.

Protocols

The protocol is also visible in the packet and is useful for routing and security purposes. For example, the filter may pass traffic in the SMTP protocol to pass to the mail server, while not allowing other IP traffic addressed to the same service to pass. Because the intent of the traffic is more obvious in the higher-level protocols, filtering by protocol can be very effective and useful.

Firewalls

It is beyond the scope of this article to provide instruction on how to build or even to operate a firewall. Within the allotted space, it is difficult to simply convey an understanding of their nature and use. A basic definition and discussion follows.

The American Heritage Dictionary defines a firewall as “a fireproof wall used as a barrier to prevent the spread of a fire.” By analogy, a network firewall is a traffic-proof barrier used to prevent the spread of disorderly or malicious traffic. More specifically, a firewall is a special collection of hardware and software that connects two networks and that is used to protect each of the assumptions as to which side of the firewall a fire will start on.

Like most analogies, this one is instructive even at the extremes where it begins to break down. In the analogy, a firewall is assumed to resist fire equally in both directions. It is symmetric; it does not have to treat fire on one side of the wall differently from fire on the other. It must resist fire, but it must pass people. However, it is easy to distinguish people from fire, and all people and all fire, on either side of the wall, are treated the same. The task of the network firewall is to distinguish between threatening and nonthreatening traffic and to do so differently depending on which side the traffic originates. In the presence of fire, a firewall need not pass people; resisting fire is more important than passing people. However, the network firewall will rarely be permitted to reject all traffic in the name of rejecting all attack traffic. It will usually be required to pass legitimate traffic, even in the presence of known attack traffic.

Moreover, a firewall is not a box; it is not a product that can be purchased off the shelf. At time of this writing, more than 40 vendors offer products that are described, at least in part, as firewalls. Although similarities among them exist, there are also fundamental differences in their approaches. Even given a complete understanding of company requirements and security policy, gaining sufficient knowledge about tens of products to decide which one is most appropriate is a major challenge.

Firewall Policy Positions

Four fundamental policy positions are available to network operators. The firewall policy will be the result of these postures and of the applications on the network.
Paranoid.
The first of these positions is called paranoid. It is motivated by extreme caution and probably fear, and characterized by the absence of a connection to the Internet.

Prudent.
The second position is called prudent or restrictive. It too is motivated by caution, but also by a recognition of the value of connection to the Internet. It is characterized by the fact that everything which is not explicitly allowed is implicitly forbidden. For example, a private Internet user would have to be explicitly authorized to Telnet to a system on the public Internet.

Permissive.
The permissive posture is the opposite of the restrictive policy. Under this policy, everything that is not explicitly forbidden is implicitly allowed. Obviously, it is the intent of this policy to forbid the necessary conditions for all known attacks. This policy is intended to provide a level of protection with a minimum of interference with applications. This is the policy most likely to be applied when applying a firewall to an existing connection. It is particularly useful if little is known about the applications and if there is a strong desire not to interfere with or break those applications. It is the policy most likely to be recommended by Internet service providers who are motivated to maximize the value of the connection.

Promiscuous.
The promiscuous policy is that anything goes. Under this policy, there are multiple connections and any legitimate packet can flow from any source to any destination.

Choosing a Firewall Policy
An interesting question is why anyone would want to be in postures one or four? Remarkably, position one is the default position for business. Most businesses have not yet connected to the Internet. Position four is the default policy for the Internet; all connections and traffic are tolerated in the name of maximizing the bandwidth and the potential for getting messages through.

If an Internet service provider is asked for guidance on a firewall policy, it will likely recommend that the position should be on the promiscuous side of permissive. The service provider will supply a list of restrictions to address all of the attacks that it knows about. However, this permits exposure to a large set of fundamental vulnerabilities. This is, in part, because the Internet service provider believes in the value of the net and does not wish to deny its clients any benefits without necessity.

This author recommends a position on the paranoid side of prudent or restrictive. In other words, permit only that traffic that is associated with a particular value for which the net is being used. The flow of all other traffic should be resisted.

A Conservative Firewall Policy
A conservative firewall policy is intended to position an institution or network on the paranoid side of restrictive. The intent is to protect not only against known and expected attacks, but also against those that have not been invented yet. It is driven by fundamental vulnerabilities, rather than by known threats and attacks. It attempts to take only those risks that are necessary to accommodate the intended applications.
In addition, no information about the private network should be available on the public net. Private net addresses should never appear on the public net; they should be replaced or aliased to an address that the firewall owns. Addresses on packets and messages should be re-encoded at the firewall. Similarly, users' internal E-mail addresses should not appear on the public net. These private addresses should be replaced with the name of the site or enterprise at the firewall on the way out and replaced on the way in.

Protocols should not traverse the firewall. Traffic should be decoded and re-encoded at the firewall. For example, a SMTP carrying a message should be decoded into a message and then re-encoded into another SMTP for transmission at the firewall.

Reusable passwords should not traverse the firewall in either direction. Incoming passwords may be replays and are not reliable evidence of the identity of the user. Outgoing passwords may be similar to those used by users on the inside, and their use across the firewall may compromise internal systems. A preference for Secure telnet or File Transfer Protocol should be made. These protocols provide end-to-encryption for all traffic, including the password. Alternatively, one-time passwords (e.g., SecureID or s-key) could be used. Although these do not protect all traffic, they protect against replays.

Proxies should represent the public net to the private net. For example, when a user of the private net wishes to access a World Wide Web (WWW) server on the public net, he or she should be transparently routed through the WWW proxy on the firewall. This proxy should hide the user's address from the public net, and protects both nets and the user. The user cannot misrepresent his or her address to the public net, and a process on the public net can directly attack only the proxy, not the user.

Only a limited set of limited applications should be permitted. Under this policy, such a limited application as E-mail is permitted, and such a very general application as telnet is discouraged. telnet is very general, flexible, and its intent is not obvious. It is vulnerable as a target and useful for attack.

Only those public applications that are intended for use on the public net should be placed on the public net. The public should not be permitted to traverse a firewall simply for the purpose of gaining access to public applications.

Applications on the public net should be implemented on dedicated and isolated servers. The server should be dedicated to a single use; it should not rely on the operating system to protect the application. Public servers should not know about the private net. Any connection to the private net should be to an application and over a trusted path. Privileged access to such servers should require strong authentication.

The public should not be granted read and write access to the same resource. For example, if the public can read a web page, they should not be able to write to it. The ability to write to it would permit them to alter or contaminate the data in a manner that could prove embarrassing. If a directory is provided to which the public can send files, they should not be able to read from that directory. If they can both read and write to the directory, they may use it simply as storage in lieu of their own. They may also use it to store contraband data that they would not want on their own systems and which might also prove embarrassing.

**Encryption**

Encryption is the application and use of secret, as opposed to public, codes. It is a powerful defense that can deal with many of the problems related to vulnerable links and even some of those related to insecure nodes. It is inexpensive and effective. In addition, multiple implementations are available. However, it is limited in the open node problems that it can deal with and may require some management infrastructure. Exhibit 1 displays some of the encryption choices available for selected applications on the Internet.
Encryption on the Internet*

<table>
<thead>
<tr>
<th>Application</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail</td>
<td>PGP, SecureXchange, PEM, S-MIME</td>
</tr>
<tr>
<td>File</td>
<td>PGP, RSA Secure, Entrust</td>
</tr>
<tr>
<td>Application</td>
<td>DES, IDEA, stelnet, sftp</td>
</tr>
<tr>
<td>Client/Server</td>
<td>Secure Socket Layer (SSL)</td>
</tr>
<tr>
<td>Gateway-to-gateway</td>
<td>Digital, IBM, TIS</td>
</tr>
<tr>
<td>World Wide Web</td>
<td>s-http</td>
</tr>
<tr>
<td>Secure IP</td>
<td>S/WAN</td>
</tr>
</tbody>
</table>

*See http://www.rsa.com for a list of products and vendors.*

Encryption is used for two fundamental purposes on the net. The first is to preserve necessary confidentiality on the net, which is the traditional use of cryptography. The second is to enable some confidence about with whom one is talking. In other words, if conversation is in a language that can only be spoken by one other, the correct parties are speaking to one another.

Encryption can also be used to resist password grabbers and other eavesdropping attacks.

Using the Internet in a Relatively Safe Environment

The following are recommendations for using the Internet in a relatively safe way. Although few will follow all of these recommendations, there is risk involved in any deviation from the recommendations. Moreover, although complete adherence to these recommendations will not eliminate all vulnerabilities, it will address many of them. Finally, although complete adherence will not eliminate all risks, following these recommendations provides a reasonable balance between risk and other values.

- **Do not rely on the secrecy or authenticity of any information traversing the internet in public codes.** Names and addresses, credit card numbers, passwords, and other data received from the public net may be replays rather than originals. Amounts and account numbers may have been tampered with.

- **Choose a single point of connection to the Internet.** Although the Internet is inherently mesh connected, and more than one connection may be necessary to avoid single points of failure, the more connections, the more points of attack and the more difficult it is to maintain consistent controls. The fewer the number or points of connection, the fewer the potential points of attack and the easier to maintain control.
- **Connect to the Internet only with equipment dedicated to that purpose.** When computers were expensive, it was economic to put as many applications as possible on the costly hardware. Communication software was added to connect existing multi-use, multi-user systems to the net. Attacks exploited this gratuitous generality. Because of less expensive hardware, hardware connected to the net should be dedicated to that use. All other applications should be run on other systems.

- **Choose application-only connections.** Many of the compromises of the Internet have resulted from the fact that the components were connected at the system layer and that attacks have succeeded in escaping the application to the more general and flexible system layer. If in an attack encounters the E-mail service, it should see nothing else. If it escapes the E-mail application, it should see nothing. Under no circumstances, should it see the prompt of an operating system that knows about any other system. In other words, the operating system should be hidden from the public net.

- **Limit the use of Telnet.** Telnet, particularly to the operating system, is a very general and flexible capability. It can be both used for attack and is vulnerable to attacks. Most of its functions and capabilities can be accomplished with safer alternatives.

- **Use end-to-end encryption for commercial applications on the net.** Although most of the applications and traffic on the public net are public, commercial and other private applications on the public net must be conducted in secret codes.

- **Require strong authentication.** Users of private applications on the public net or of the public net for commercial applications must use strong authentication. Two independent kinds of evidence should be employed to determine the identity of a user, and the authentication data must be protected from capture and replay.

- **Log, monitor, and meter events and traffic.** Given enough time, almost any attack can succeed. It is important to be able to recognize attack traffic and correct for it early. Attacks can usually be recognized by a change, often a sudden increase, from normal traffic patterns. It is useful to know what normal traffic looks like to be able to recognize variances on a timely basis, and to communicate the condition of those variances to managers who can take timely corrective action.

**Conclusion**

The Internet is as ubiquitous as the telephone and for similar reasons. It gives users such an economic advantage over nonusers so that the nonusers are forced to become users. Pundits are fond of saying that no one is making money on the Internet. This position is fatuous and suggests that tens of thousands of enterprises are behaving irrationally. What is meant is that no one is conducting commerce on the Internet, at least not in the sense that they are selling, distributing, billing, and being paid over the Internet. Of course, many firms are doing one or more of these. Many others are making money, mostly by reducing costs. Many companies are using the Internet because it is the most efficient way to support customers.

The Internet holds out the promise to empower, enrich, and perhaps even ennoble. A minimum level of public trust and confidence must be maintained if that promise becomes a reality. That trust is both fragile and irreparable.

Because fundamental vulnerabilities on the network exist and because all possible attacks cannot be anticipated, a conservative policy and a responsive posture are required.
Author Biographies

William Hugh Murray
William Hugh Murray is an Executive Consultant in IS Security for Deloitte & Touche.