Payoff
Firewalls are an excellent security mechanism to protect networks from intruders, and they can establish a relatively secure barrier between a system and the external environment. Firewall products are available with a variety of functionality and features, such as strong authentication, the ability to create VPNs, and easy-to-use interfaces. Selecting the appropriate firewall for an organization's needs requires careful consideration of the available types and products.

Introduction
Firewalls are an excellent security mechanism and, when appropriately selected and implemented, can establish a relatively secure barrier between a system and the external environment. This article describes the types of firewalls that are available and presents the advantages and disadvantages of each type.

Packet Filters
The most basic type of firewall is a packet filter. It receives packets and evaluates them according to a set of rules that are usually in the form of access control lists. These packets may be forwarded to their destinations, dropped, or dropped with a return message to the originator describing what happened. The types of filtering rules vary from one vendor's product to another, but those most frequently applied are:

- Source and destination IP address (e.g., all packets from source address 128.44.9.0 through 128.44.9.255 might be accepted, but all other packets might be rejected).
- Source and destination port (e.g., all TCP packets originating from or destined to port 25—the simple mail transfer protocol, or SMTP, port—might be accepted, but all TCP packets destined for port 79—the finger port—might be dropped).
- Direction of traffic (e.g., inbound or outbound).
- Type of protocol (e.g., IP, TCP, user datagram protocol, or internetwork packet exchange).
- The packet's state (i.e., SYN, meaning synchronize, or ACK, which is the acknowledgement that a connection between hosts has already been established).

Packet-filtering firewalls provide a reasonable amount of protection for a network with minimum complications. Packet-filtering rules can be extremely intuitive and thus easy to set up. One simple, but surprisingly effective, rule is to allow all packets that are sent from a specific, known set of Internet protocol (IP) addresses, such as hosts within another network owned by the same organization or corporation. Packet-filtering firewalls also tend to have the least negative effect on the throughput rate at the gateway compared with other types of firewalls. They also tend to be the most transparent to legitimate users. If the filtering rules are set up appropriately, users obtain their required access with little interference from the firewall.

However, simplicity has its disadvantages. The rules that packet-filtering firewalls implement are based on port conventions. If an organization wants to stop certain service
requests (e.g., telnet) from reaching internal or external hosts, the most logical rule is to block the port (e.g., port 23) that by convention is used for telnet traffic. Blocking this port, however, does not prevent someone inside the network from allowing telnet requests on a different port that the firewall's rules leave open. In addition, blocking some kinds of traffic causes a number of practical problems. Blocking X-Windows traffic (which is typically sent to ports 6000 to 6013), on the surface, seems to provide an effective security solution because of the many known vulnerabilities in this protocol. Many types of remote log-on requests and graphical applications depend on X-Windows, however. Blocking X-Windows traffic may thus restrict functionality, leading to the decision to allow all X-Windows traffic (which makes the firewall a less than effective security barrier).

In short, firewalling schemes based on ports do not provide the precise control that many organizations require. Moreover, packet-filtering firewalls are often deficient in logging capabilities, particularly in providing logging that can be configured to an organization's needs (e.g., to capture only certain events in some cases and, in others, to capture all events). They may also lack remote administration facilities that can save considerable time and effort. Finally, the process of creating and updating filtering rules is prone to logic errors that could result in easy conduits of unauthorized access to a network.

Like most other security-related tools, over time, many packet-filtering firewalls have become more sophisticated. Some vendors of packet-filtering firewalls offer programs that check the logic of filtering rules to discover any contradictions and errors. Packet-filtering firewalls also exist that offer strong authentication mechanisms, such as token-based authentication. Many products defend against previously successful methods to defeat packet-filtering firewalls. However, network attackers can send packets to or from a disallowed address or disallowed port by fragmenting the contents. Fragmented packets cannot be analyzed by a conventional packet-filtering firewall, so the firewall allows them through, where they are assembled at the destination host. In this manner, the network attackers can bypass firewall defenses.

Similarly, attackers have developed an attack in which the initial TCP packet sent has the acknowledge (ACK) rather than the synchronize (SYN) flag set in the header. Because SYN packets are used to establish connections, a conventional packet-filtering firewall is programmed, for efficiency, to analyze and apply filtering rules only to SYN packets. ACK packets can freely pass through the choke because they should be part of an established connection. However, some vendors have developed a state-conscious firewall, which is a special type of packet-filtering firewall, to prevent these types of attacks. By remembering the state of connections that pass through the firewall, a state-conscious firewall can prevent packet fragmentation and ACK packet-based attacks. Some state-conscious firewalls can associate each outbound connection with a specific inbound connection (and vice-versa), creating simpler rules. Because the user datagram protocol (UDP) is connectionless and thus does not contain information about states, state-conscious firewalls are vulnerable to UDP-based attacks unless they track each UDP packet that has already gone through and determine which subsequent UDP packet sent in the opposite direction (i.e., inbound or outbound) is associated with that packet.

Many routers have packet-filtering capabilities and, therefore, can be considered a type of firewall. Using a packet-filtering router as the sole choke component within a gate, however, is not likely to provide effective security, because routers are more vulnerable to attack than firewall hosts are, and they generally do not log traffic efficiently. A screening router is difficult to administer, often requiring that a network administrator download its configuration files, edit them, and send them back to the router. The main advantage of screening routers is that they provide some filtering functionality with minor performance overhead and minimal interference to users, who may not realize the screening router's existence. One option for using packet-filtering routers is to employ them as external routers in a belt-and-suspenders topology. The security filtering by the external router provides additional protection for the firewall by making unauthorized access more difficult. Moreover, the gate has more than one choke component, providing multiple
barriers to prevent an attack on an internal network and to compensate for configuration errors and vulnerabilities in any one of the choke components.

**Application-Gateway Firewalls**

A second type of firewall handles the choke function of a firewall differently—by determining not only whether but how each connection through it is made. This type of firewall stops each incoming (or outgoing) connection at the firewall, and, if the connection is permitted, initiates its connection to the destination host on behalf of whoever created the initial connection. This type of connection is called a proxy connection. By using its data base, which defines the types of connections allowed, the firewall either establishes another connection (i.e., permitting the originating and destination host to communicate) or drops the original connection. If the firewall is programmed appropriately, the process can be transparent to users.

An application-gateway firewall is simply a type of proxy server that provides proxies for specific applications. The most common implementations of application-gateway firewalls provide proxy services, such as mail, file transfer protocol (FTP), and telnet, so that they do not run on the actual firewall, which increases security. Each connection is subject to a set of specific rules and conditions similar to those in packet-filtering firewalls, except that the selectivity rules application-gateway firewalls use are not based on ports, but on the to-be-accessed programs or services (regardless of which port is used to access these programs). Criteria, such as the source or destination Internet protocol (IP) address, however, can be used to accept or reject incoming connections. Application-level firewalls also determine permissible conditions and events when a proxy connection has been established. An FTP proxy can restrict FTP access to one or more hosts by allowing the get command, for example, and, at the same time, preventing the put command. A telnet proxy can terminate a connection if the user attempts to perform a shell escape or to gain root access. Application-gateway firewalls are not limited only to applications that support TCP/IP services, however. These tools can similarly govern conditions of usage for a variety of applications, such as financial or process control applications.

The two basic types of application-gateway firewalls are: application-generic firewalls and application-specific firewalls. The application-generic type provides a uniform method of connection for every application, regardless of type. The application-specific firewall determines the nature of connections to applications on an application-by-application basis. Regardless of the type of application-gateway firewall, if properly configured, the resulting security control can be precise—considerably more than is possible with packet-filtering firewalls. If used in connection with appropriate host-level controls (e.g., proper file permissions and ownerships), application-gateway firewalls make externally originated attacks on applications difficult. Another important function of application-gateway firewalls is hiding information about hosts within the internal network from the rest of the world. Finally, a number of commercial application-gateway firewalls are available that support strong authentication methods, such as token-based methods (e.g., use of hand-held authentication devices).

Application-gateway firewalls are the best selling of all types of firewalls. Nevertheless, they have some notable limitations. Most significant, for every TCP/IP client for which the firewall provides proxies, the client must be aware of the proxy that the firewall runs on its behalf. Therefore, each client must be modified accordingly. A second limitation is that, unless one uses a generic proxy mechanism, every application needs its own custom proxy. In the case of proxies for widely used services, such as telnet, FTP, and hyper text transfer protocol (HTTP), this limitation is not formidable because a variety of proxy implementations is available. Proxies for many other services are not available at the present time, however, and must be custom-written. Although some application-gateway firewall implementations are more transparent to users than others, none are
completely transparent. Some require users, who have initiated connections, to make selections from menus before they reach their destinations. Finally, most application-gateway firewalls are not easy to initially configure and update correctly. To use an application-gateway firewall to its maximum advantage, network administrators should set up a new proxy for every new application accessible from outside a network. Furthermore, network administrators should work with application owners to ensure that specific restrictions on usage are placed on every remote connection to each critical application from outside the network. Seldom, however, are such practices observed because of the time, effort, and complexity involved.

Circuit-Gateways Firewalls

As discussed previously, application-gateway firewalls receive connections from clients, dropping some and accepting others, but always creating a new connection with whatever restrictions exist whenever a connection is accepted. Although, in theory, this process should be transparent to users, in practice, the transparency is less than ideal. A third type of firewall, the circuit-gateway firewall, has been designed to remedy this limitation by producing a more seamless, transparent connection between clients and destinations using routines in special libraries. The connection is often described as a virtual circuit, because the proxy creates an end-to-end connection between the client and the destination application. An application-gateway firewall is also advantageous because, rather than simply relaying packets by creating a second connection for each allowed incoming connection, it allows multiple clients to connect to multiple applications within an internal network.

Most circuit-gateway firewalls are implemented using SOCKS, a tool that includes a set of client libraries for proxy interfaces with clients. SOCKS receives an incoming connection from clients, and if the connections are allowed, it provides the data necessary for each client to connect to the application. Each client then invokes a set of commands to the gateway. The circuit-gateway firewall imposes all predefined restrictions, such as the particular commands that can be executed, and establishes a connection to the destination on the client's behalf. To users, this process appears transparent.

As with application-gateway firewalls, circuit-gateway firewall clients must generally be modified to be able to interface with the proxy mechanism that is used. Making each client aware of SOCKS is not an overwhelming task, because a variety of SOCKS libraries are available for different platforms. The client must simply be compiled with the appropriate set of SOCKS libraries for the particular platform (e.g., UNIX or Windows) on which the client runs.

Circuit-gateway firewalls also have limitations. First and foremost, the task of modifying all clients to make them aware of the proxy mechanism is potentially costly and time-consuming. Having a common interface to the proxy server so that each client would not have to be changed would greatly improve usability. Second, circuit-gateway firewalls tend to provide a generic access mechanism that is independent of the semantics of destination applications. Because, in many instances, the danger associated with specific user actions is dependent on each application, offering proxies that take into account application semantics would be more advantageous. (Invoking the delete command to delete data in an application that reinitializes all parameter values by retrieving values from a data base that is not accessible to users every time it is invoked, for example, is potentially not catastrophic. In other applications, however, being able to delete data is potentially hazardous.) In addition, SOCKS has several limitations. Most implementations of SOCKS are deficient in their ability to log events. Furthermore, SOCKS neither supports strong access authentication methods nor provides an interface to authentication services that could provide this function.
Hybrid Firewalls

Although the distinctions between packet-filtering firewalls, application-gateway firewalls, and circuit-gateway firewalls are meaningful, many firewall products cannot be classified as exactly one type. For example, one of the most popular firewall products on the market is basically a packet-filtering firewall that supports proxies for two commonly used TCP/IP services. As firewalls evolve, it is likely that some of the features in application-gateway firewalls will be included in circuit-gateway firewalls, and vice-versa.

Virtual Private Networks

An increasingly popular Internet security control measure is the virtual private network (VPN), which incorporates end-to-end encryption into the network, enabling a secure connection to be established from any individual machine to any other. At present, this technology is most commonly implemented in firewalls, allowing organizations to create secure tunnels across the Internet, as shown in Exhibit 1. Attackers who have planted one or more network capture devices along the route used to send packets between the firewalls do not gain any advantage from capturing these packets unless they can crack the encryption key—an unlikely feat unless an extremely short key is used. The chief disadvantage of the firewall-to-firewall VPN is that it does not provide an end-to-end tunnel. In this scheme, packets transmitted between a host and the firewall for that host are in cleartext and are thus still subject to being captured. Increasingly, however, vendors are announcing support for end-to-end VPNs, allowing host-to-host rather than only firewall-to-firewall tunnels.

A Virtual Private Network

Like any other type of Internet security control measure, VPNs are not a panacea. Anyone who can break into a machine that stores an encryption key can subvert the integrity of a VPN. VPNs do not supplant firewalls or other kinds of network security tools, but rather supplement the network security administrator's arsenal with capabilities that were not previously available. With the point-to-point tunneling protocol (PPTP) standard currently being widely implemented in VPN products (usually in firewalls with VPN support capabilities), the task of setting up secure tunnels is at least much less formidable than it had been until recently.

Conclusion

The successful use of a firewall is dependent on the selection of an appropriate product. Packet-filtering firewalls accept or deny packets based on numerous rules that depend on the source and destination ports of packets and other criteria. This type of firewall is the closest option to a plug-and-play firewall solution, although it is also generally the easiest to defeat. Proxy-based firewalls, such as circuit-gateway firewalls, are generally more difficult to defeat, and the resulting virtual circuit connection is for relatively transparent to users. However, circuit-gateway firewalls do not understand the semantics of applications and thus lack a certain amount of granularity of control. Application-gateway firewalls are also proxy-based, but connect a specific client to a specific application. Application-gateway firewalls can provide more granularity of control, but require that every application that proxies reach be modified, and they are generally less transparent to users than circuit-gateway firewalls.
Author Biographies

E. Eugene Schultz
E. Eugene Schultz, PhD, is a program manager at SRI Consulting.