computer security

by Steven W. Lodin, *Ernst & Young LLP* &
Christoph L. Schuba, *Sun Microsystems Inc*

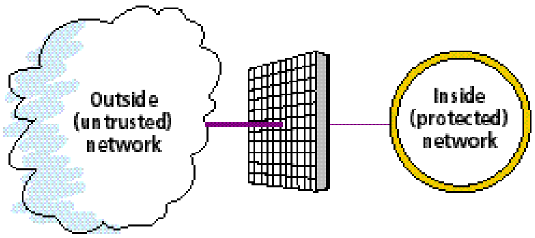
FIREWALLS FEND OFF INVASIONS FROM THE NET

Imperfect but essential, these guardians of computer network security work best if planned carefully and maintained with equal care

The first line of defense against external threats to computer systems and networks is a firewall. Whether a computer is in a corporation, government agency, university, small business, or home, if it is connected by a network to other computers, its resources, plans, and data are at risk--and so is the reputation of its owners. A firewall can help reduce that risk to an acceptable level.

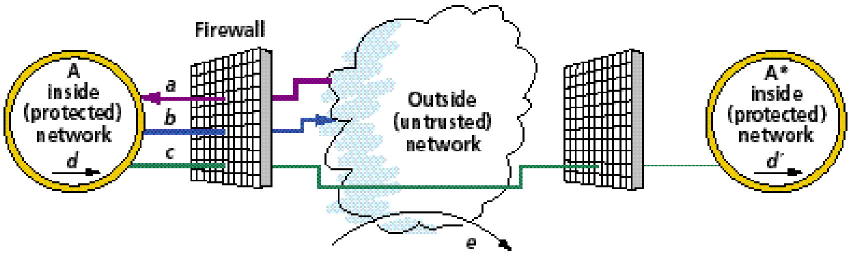
Firewall technology is a set of mechanisms that collectively enforce a security policy on communication traffic entering or leaving a guarded network domain. The security policy is the overall plan for protecting the domain. Embodied in hardware, software, or both, a firewall guards and isolates the domain [Fig. 1]. The name, of course, comes by way of analogy with a structural fire wall that blocks the spread of fire in a building.

Defining Terms



[Fig. 1] A firewall guards and isolates an inside (private) network - an intranet - from an outside (hence untrusted) network: the Internet, for instance. A firewall may also guard some parts of an internal network against other parts.

Broadly, firewalls attempt to maintain privacy and ensure the authenticity of data communications that pass through their domain’s boundaries. Whether data is entering or leaving a domain, it is protected from eavesdropping (passive wiretapping) and change (active wiretapping). But only communication traffic entering or leaving a domain comes under the influence of firewall technology. Traffic that stays inside or outside a domain is unaffected [Fig. 2].



[Fig. 2] Domains A and A*, though parts of one organization's network, are physically separate and communicate through an outside (untrusted) network. Firewalls can only control communication traffic to, from, or through that outside network--such as messages a, b, and c. They cannot control messages d and d', which do not leave the protected networks' boundaries, and connection e, which simply extends through the outside network.

Firewalls do, however, protect other material located in the interior of the domain--the stored data, computation resources, and communication resources. These are guarded against unauthorized access, browsing, leaking, modification, insertion, and deletion. And they provide a measure of protection from "denial of service," in which users inside the domain are prevented from accessing the network by a message that disables communication equipment or by a flood of messages that clogs the internal network.

One such incident made headlines in September 1996, when many Internet sites were attacked by software that made attempts to connect with them--so many attempts, in fact, that legitimate users were no longer able to access them. The perpetrator was never found, but the attack almost drove one Internet service provider out of business.

Serving as system guardians, firewalls are only part--albeit an important part--of a comprehensive network protection scheme. A balanced approach to security overall rests on well-thought-out strategies to safeguard physical plant, personnel, operations, and communications.

While firewall technology is still very much a developing technology, it is possible to explore its benefits and drawbacks and classify its components. A set of criteria can be established for evaluating firewalls as an aid in designing new firewalls or in determining whether an existing firewall is fulfilling its purpose. Clearly the current challenges to firewalls and the outlook for their future are worth exploring.

Firewall foundations

So far, firewalls have been applied to communication based on the transmission control protocol/Internet protocol (TCP/IP) that governs the public Internet and private intranets. But their concepts are applicable to other network protocols, too. Generally, firewalls provide security for distributed systems and are usually used to protect entire corporate networks.

Not everyone approves of firewalls, however, and discussions of them tend to get emotional. On the one hand, some people strongly favor firewalls because so many computer systems and networked applications simply are not secure and could use them. They argue that firewalls are more than just a retrofit patch for shortcomings in systems and protocols. What’s more, even if the host system is secure, firewalls serve as a central focus of security policy and a place to conduct comprehensive security audits.

Moreover, firewalls address some of the problems of network security that security mechanisms in a host do not. Because security functions are combined and concentrated in a firewall, proponents say, installing, configuring, and managing them is simpler. They also make the administration of the system and the management of the network more efficient because they are transparent to users, limit exposure of the internal network, and can accommodate almost any internal network topology. Also, they are widely accepted, readily available, and easily justifiable to purchasing managers.

On the other hand, opponents claim that firewalls generate a false sense of security that leads to laxity in enforcing security measures. This objection was expressed in a now popular analogy to a candy bar: firewalls provide" a hard, crunchy outside with a soft, chewy center" (RFC 1636). [See To Probe Further, for more on such requests for comments (RFCs) and informational bulletins (FYIs), posted on the World Wide Web].

Firewall development typically diverts resources away from improving the security of the computer system behind the firewall. Other criticisms are that firewalls do not provide perfect security; they offer no protection against malicious insiders, for example, and they fail to protect against any connection activities that circumvent them. People can use unauthorized modems attached to computers inside the firewall. They also offer only limited protection against illicit rendezvous (unauthorized connections) and "data-driven" attacks, such as those carried out by malicious executable code in apparently innocent downloaded Java applets or ActiveX controls. But perhaps their most serious limitation is that firewall designers have concentrated so far on acute problems at hand. The reactive nature of firewall design, opponents believe, means that firewalls could be vulnerable to novel attack scenarios of the future.

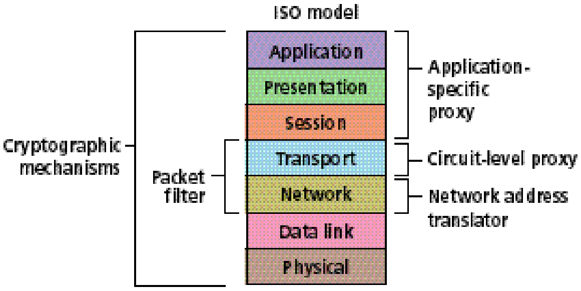
Nor are firewalls free of operational difficulties. For instance, changes in the firewall system configuration may produce security holes (because, in the usual current practice, the actual configuration is not checked against the security policy). Or the protection mechanisms implemented in the firewall may actually work against each other.

All of these are valid points. Firewalls *do* create a focus for security--but that is a critical advantage. And they *do* suffer from a variety of problems--which, however, are solvable or at least controllable by security policy measures, technological development, and, equally important, theoretical development.

Shortcomings notwithstanding, most computer security experts would agree that a firewall is an important part of security which, in these dangerous times, offers major benefits. In short, a firewall is one of the best available ways to enforce a security policy and keep a site secure.

Defense mechanisms

The various security mechanisms that firewalls employ correspond roughly to certain layers in the open systems interconnection (OSI) model of networking established by ISO the international standardization organization [Fig. 3]. The packet-filtering mechanism, for example, operates primarily on the network and transport layers, while the network address translation mechanism operates solely on the network layer. Operating on the transport level is the circuit-level proxy mechanism while the application-specific proxy mechanism operates on all three top levels.



[Fig. 3] In general, the various firewall security mechanisms address themselves to specific layers in the open systems interconnection (OSI) network model. Several mechanisms can be combined into a comprehensive firewall system, but the mechanisms should be chosen and coordinated so that they do not work against each other.

At all seven layers, cryptographic mechanisms can be applied. For example, at the transport layer, they can provide end-to-end privacy of communications transparently to the user; at the application layer, they can provide application-specific user authentication.

Firewall mechanisms do impose overhead in the form of decreased data throughput, which means, of course, increased delay. But over the years, performance has continuously improved as a direct result of higher processor speeds and software code optimizations.

In general, the lower the mechanism operates in the OSI layers, the higher its throughput and the lower its delay. Conversely, the stronger the cryptographic mechanism or the more complex the filtering rules, the lower the throughput and the higher the delay.

Checking each packet

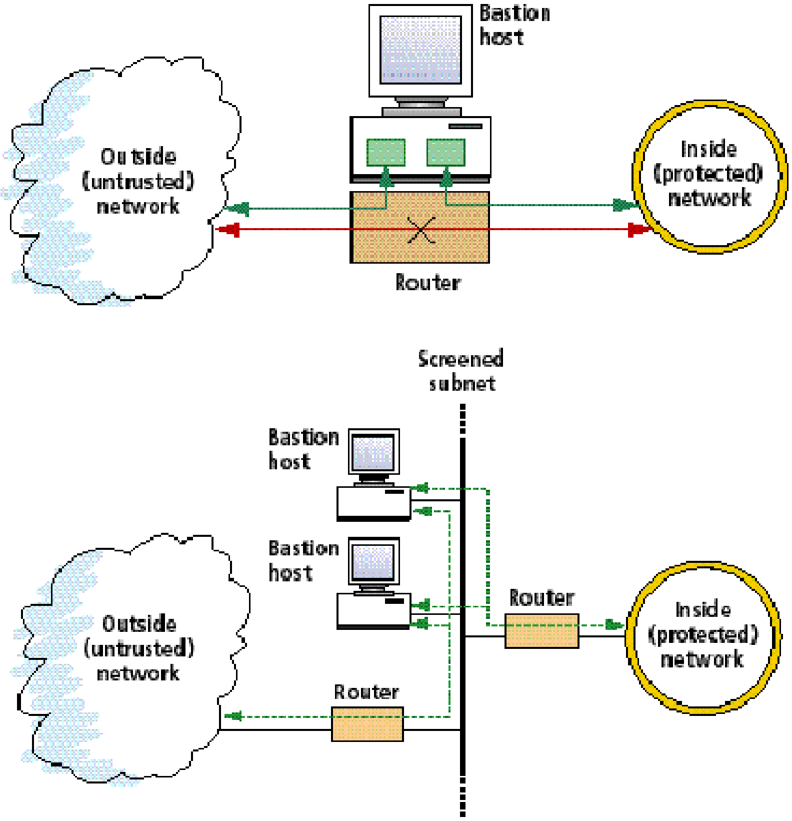
In packet-filtering, a router either allows or denies the passage of data after checking its header and contents for conformance
to a set of rules that reflect a security policy. In a TCP/IP packet-filtering firewall, the router subjects arriving datagrams (a data packet) to a filtering mechanism that decides whether to forward or discard each datagram. The filter usually examines the source and destination addresses and protocol port numbers contained in the header. The rules operate on datagrams individually, without regard to state information (data on what has previously occurred in the system), so they are viewed separately from other datagrams that are part of the same connection.

A variation on this stateless filtering, called stateful packet-filtering, saves state information for packets that belong to the same connection so that decisions can be made within the context of a particular message. Inbound file transfer protocol (ftp) data connections, for example, can thus be selectively allowed through a firewall if authorized by an ftp control connection. Before stateful packet-filtering, all ftp data connections were allowed through the firewall, which opened networks up to several kinds of attacks.

Packet-filtering gives security personnel the opportunity to examine and verify all data passing through the firewall. But it does not establish a true security association, and the integrity and authenticity of the packets examined cannot be controlled. (In RFC 1825, a security association is defined as "the set of security information relating to a given network connection or a set of connections.")

Verifying each packet naturally induces some delay and jitter, but commercial filtering software designed to minimize delays is available. A more serious objection to verification is the difficulty of developing a set of filtering rules (in low-level specification language) from a statement of security policy (in high-level human language). Even with automated rule-writing tools, graphical user-interfaces, and much experience, generating the accept/reject rules for a complex security policy is a challenge.

Two important examples of packet-filtering firewalls are the screened-host firewall and the screened-subnet firewall. In the first, a router controls access to and from a single host. In the second, a pair of routers protect a "demilitarized zone" network, also referred to as a screened subnet, consisting of one or more bastion hosts [Fig. 4].


Bastion host
Outside (untrusted) network
Inside (protected) network
Router
Screened subnet
Bastion host
Bastion host
Router
Outside (untrusted) network
Inside (protected) network
Router

[Fig. 4] All packet-filter firewalls deny access to traffic that does not meet a set of rules [indicated by a red line with x] and pass traffic that does [green lines with arrowheads].

In a screened-host firewall [top], a router at network level controls access to and from a single host--called a bastion host--through which all traffic to and from the protected network must travel. Direct access to the protected network is denied and the bastion

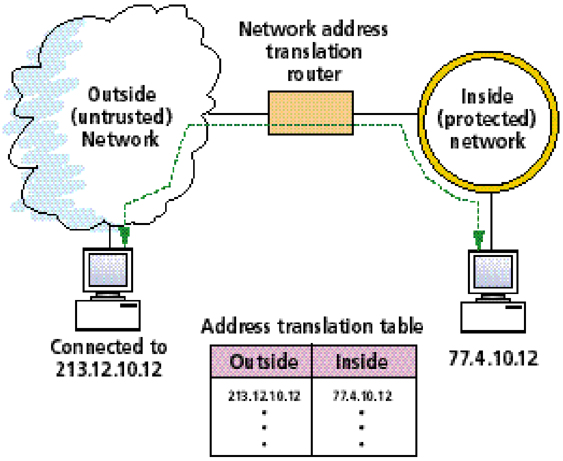
host does not forward packets. The bastion host is a highly defended, secured strongpoint that--one hopes--can resist attack.

In a screened-subnet firewall [above], a pair of routers control access to a small network of bastion hosts. The screened subnet is also called a "demilitarized zone" (DMZ).

Changing addresses

Originally the network address translation (NAT) mechanism was proposed as a short-term solution to the growing shortage of Internet protocol (IP) addresses, not as a security mechanism (RFC 1631). But happily for security administrators, the NAT hides the internal addresses and network topology of its protected domain from the outside. Such obscurity thwarts several methods of attack because little or no target information is available to the outside. Furthermore, hardly any network-based services, except for address translation, connect directly to the outside and therefore the domain is less susceptible to attack than if it were guarded by ordinary routers.

Network address translation devices are placed at the borders of network domains [Fig. 5]. Each NAT device contains a table of address pairs: the local IP address and the corresponding globally unique address. For all outgoing datagrams, the device translates the local address into its associated global address, and for all incoming packets, it translates the global address into its local address.


Network address translation router
Outside (untrusted) Network
Inside (protected) network
Connected to 213.12.10.12
77.4.10.12
Address translation table

Outside	Inside
213.12.10.12	77.4.10.12
⋮	⋮

[Fig. 5] A network address translator hides internal addresses from the outside world. Network address translation (NAT) routers contain a table of outside and inside addresses. They translate the outside address of an incoming message into the hidden inside address, and do the reverse for an outgoing message.

The address association can be static (not changing, once administratively assigned) or dynamic (the externally visible addresses are taken from a pool of addresses and are automatically assigned to internal addresses on a need basis). The NAT discards the end-to-end significance of addresses, making up for the loss with increased state information in the network.

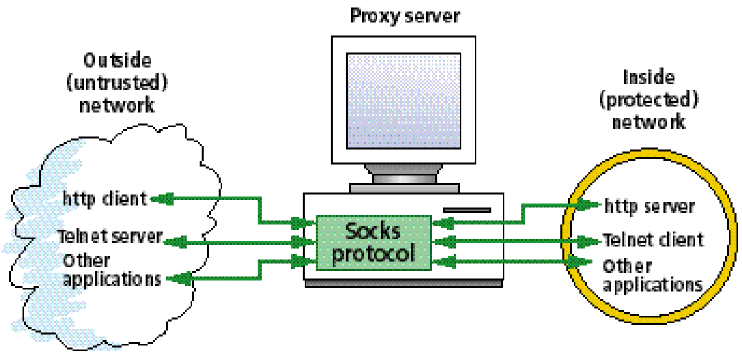
Like any router, a NAT router keeps messages separate when two inside users send to the same outside destination (an Internet service provider, say, or a much-used Web site) or when two outside sources direct simultaneous messages to the same inside user--that is, the NAT maintains session association. The only difference is that the NAT changes addresses belonging to hosts inside the protected network so that they appear different on the outside.

The NAT does not work if higher-layer protocols or applications use and expose the hidden local address. An example is domain name system (DNS) messages; DNS is an Internet protocol and widely distributed database that provides binding resolutions between host names and IP addresses. In some cases, such as electronic mail, though, NAT devices can prevent this problem by rewriting higher-layer-protocol messages with appropriately mapped addresses. Applications that carry and use local addresses across a NAT boundary will not work unless the NAT device is able to detect the addresses imbedded in the packets and correctly translate them.

Circuit-level forwarding

Firewalls that use the circuit-level forwarding mechanism group packets into connections--for example, TCP connections--by maintaining state information across the packets. For example, the firewall may insert a generic transport-layer proxy process into the connection. Inbound as well as outbound connections must connect to the proxy process before they can proceed further. To determine whether the connection should be established or blocked, the proxy makes use of access rules, which can be elaborate, and can require authentication and additional client/proxy protocol message exchanges.

One of the most popular examples of this mechanism is Socks (RFCs 1928, 1929, and 1961) [Fig. 6], which has become the *de facto* standard for proxying on the Internet. Software for Socks firewalls is widely available, and many client programs, both commercial and free, have Socks support built into them.



[Fig. 6] Many firewalls now include built-in support for Socks (the name derives from Unix Sockets), software that allows applications to access a variety of communication protocols. Thus Socks can handle many different types of traffic, routing packets between compatible clients and servers in the untrusted network and the protected one. In effect, it forms a circuit between a client and server; but it acts as a proxy, too, forwarding only those packets deemed acceptable.

The generic circuit-level forwarding mechanism can be hidden in low-level libraries; no modification of client/server source code is needed if user interaction is not required. Programs that initiate connections, however, may need to be modified so that they can provide authentication information. Only a few changes are necessary, but there are challenges, such as ensuring the availability of source code, coping with the heterogeneity of system platforms, distributing programs, and educating all of the potential users.

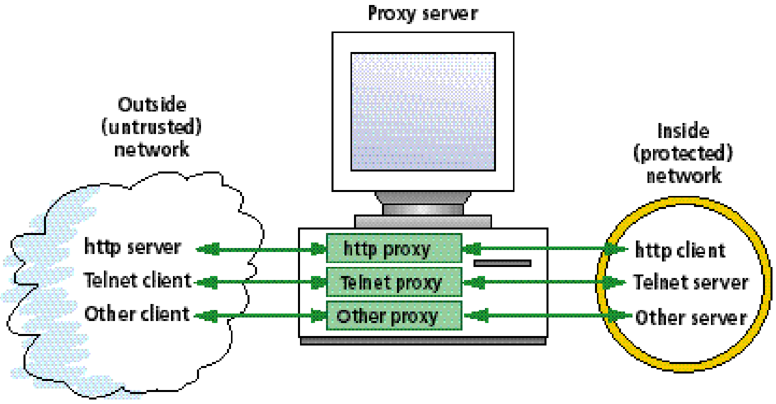
Application-level forwarding

In using the application-level proxy forwarding mechanism, firewalls can interpret the data contained in the packets that make up a communication in accordance with the particular application protocols and make security decisions on the basis of that interpretation. This mechanism is similar to circuit-level proxy forwarding, but it can also interact with the communication end points through the application's high-level protocol. It should be noted that this process is, by its nature, application-specific, so for each application, a new forwarding service must be provided.

Among the application-level proxy's many advantages are the fine-grained authentication and access control it offers by virtue of the fact that it interprets the application protocol in use. Also, the mechanism is implemented using small programs that can be more easily scrutinized for vulnerability before deployment than can ordinary server programs. And, if the mechanism does fail, it provides a single point at which the cause can be investigated. An example of a set of software tools and implementation practices for building servers according to the application-specific philosophy is the TIS Internet Firewall Toolkit from Trusted Information Systems (TIS) Inc., Glenwood, Md. (http://www.tis.com/).

But application-specific forwarding has a few drawbacks as well. Like generic circuit-level forwarding, it requires some modification of client software. And it carries protocol processing overhead that reduces its efficiency; it must make two additional round trips through the protocol stack as it interacts with the end points.

The "dual-homed" gateway [Fig. 7] is a representative application-level firewall. Like the bastion host, it generally has two independent network interfaces--one on the external network and one on the internal (protected) network. To insure isolation of the two networks, it does not automatically forward traffic between the two interfaces.



[Fig. 7] An application-level firewall uses application-specific proxies that can interact

with the source and destination of a message to determine whether it meets security standards, and then allows or denies access on the basis of its evaluation. Separate proxies are needed for each application. Further, a so-called "dual-homed" application-level firewall can be built by installing two interfaces, one on each network. So a popular location for such a firewall is a bastion host, in either a screened-host or screened-subnet firewall [see Fig. 4].

Cryptographic mechanisms

Security mechanisms based on the enciphering and deciphering of messages in secret code are available in great variety. An example is the authentication header (AH) mechanism, which ensures integrity and provides authentication without maintaining confidentiality (RFC 1826). Another is the encapsulating security payload (ESP) mechanism, which guarantees confidentiality and, optionally, integrity and authentication (RFC 1827). Both mechanisms can operate between a set of hosts and/or gateways--that is, end to end, end to intermediate, or intermediate to intermediate, in either unicast or multicast mode (single recipient or many recipients). Both were also standardized by the Internet Engineering Task Force for the IP security (IPsec) protocols.

While these mechanisms protect traffic against eavesdropping, unnoticed modification, insertion, and deletion, they do not protect against analysis of traffic. But their strength lies in their usefulness in building virtual private networks across untrusted networks like the Internet, and in enforcing a variety of security policies.

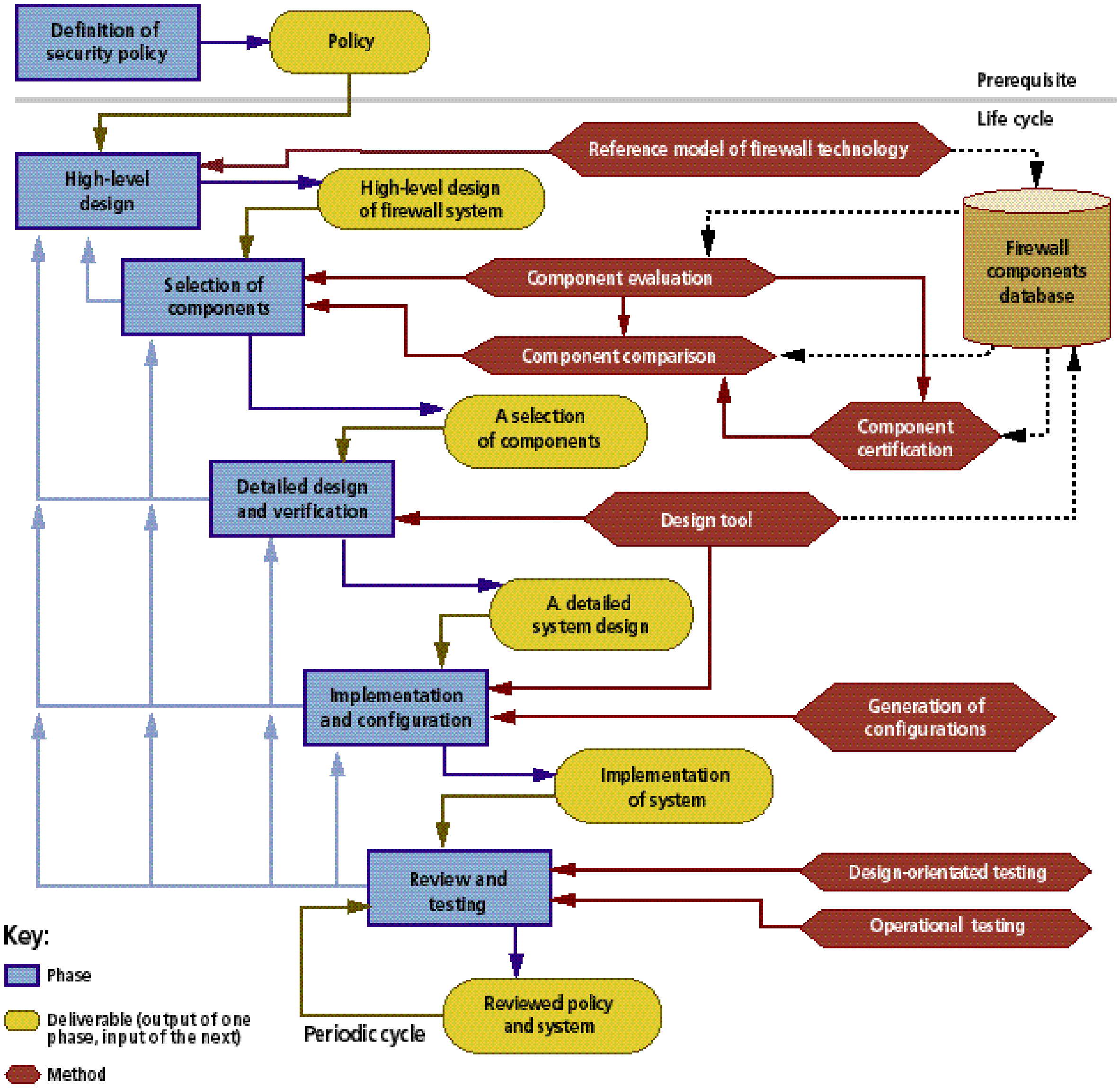
They are also algorithm-independent, although algorithm identifiers in the security protocol specify standard default algorithms. This provision ensures interoperability of AH and ESP mechanisms in all implementations.

For them to work properly, however, a security association has to be established among the set of hosts and gateways that are party to the protected communications. Of concern, too, are such practical matters as managing the distribution of encryption keys among the hosts and gateways along with minimizing the protocol processing overhead and communication delay introduced by encryption.

Incidentally, encryption, when it is combined with packet-filtering, provides a good illustration of how security mechanisms can work against each other. If the packet filter is not part of the cryptographic mechanism’s security association, it is unable to perform its function because it is unable to decrypt the higher-layer-protocol headers that it needs to enforce its filtering rules.

Life cycle of a firewall

Like software, firewall systems undergo a gradual development and evolution, called a life cycle, much like the "waterfall" life cycle of software. According to the waterfall model [Fig. 8], the first requirement is to define the network domain security policy. This is perhaps the most difficult phase of the waterfall model because of the vital questions that must be answered. What is the perimeter of the internal network being protected? Which internal services should be available to outside entities? What are those entities and which outside services should be available to users inside the protected domain? What controls are needed? What is the impact of outside services on security? What assumptions are made regarding service and system behavior? Hard as it is to develop, such a well-defined security policy is a prerequisite for the succeeding stages of the model.


Definition of security policy
Policy
Prerequisite
Life cycle
Reference model of firewall technology
High-level design
High-level design of firewall system
Firewall components database
Selection of components
Component evaluation
Component comparison
Component certification
A selection of components
Detailed design and verification
Design tool
A detailed system design
Implementation and configuration
Generation of configurations
Implementation of system
Review and testing
Design-orientated testing
Operational testing
Reviewed policy and system
Periodic cycle
Key:
Phase
Deliverable (output of one phase, input of the next)
Method

[Fig. 8] The phases of the firewall's life cycle, shown in blue rectangles, use the methods in the brown hexagonals to the right to produce the results noted in the beige ovals. The life cycle progresses diagonally, beginning with the all important definition of security policy and arriving at implementation, review, and testing after high-level design, selection of components, and detailed design. Even after the firewall is in use, periodic review and testing during the system's lifetime may result in an earlier phase being revisited (indicated by the upward-pointing blue arrows), as when a new, improved firewall component becomes available or when defects in an earlier phase are discovered.

The next phase is the decision by the firewall designer on a high-level (general or architectural) structure for the firewall system (ideally, following a reference model for the firewall technology). At this stage, there is no concern about details.

In the third stage, the designer browses through a comprehensive list of firewall products to find components that are likely to be useful, perhaps giving preference to certified products. Before a final selection is made, some unfamiliar products may need to be evaluated.

Using the selected components, the designer can then develop a low-level (detailed) structure for the firewall system. (At this stage, a design tool for firewalls, if one were available, would allow graphical design of firewall systems using descriptions of firewall components from a library, simulations of the behavior of the designed system, and formal verification of certain properties of interest. Such a tool, which is badly needed, could be used to generate configurations and software components for the firewall in its implementation and configuration phase.)

Once these details have been decided upon, the firewall system is built and deployed and the designer has an expert test it. This step ensures that the firewall is capable of enforcing the chosen security policy. Simultaneously, the security policy itself is reviewed.

After it goes into service, the firewall will be tested periodically to revalidate confidence in its installation, and corrections will be made as necessary--a form of preventive maintenance. The system will probably be subject to frequent changes in its internal configuration, network configuration, and security policy.

Checking a firewall’s security

Authors Simpson Garfinkel and Gene Spafford in their book *Practical UNIX and Internet Security* [see To probe further] note that "a computer is secure if you can depend on it and its software behaves as you expect." By this characterization, natural disasters are as much a threat to computer security as a disgruntled employee or faulty software. Clearly, determining a system’s security is hard because the absence of vulnerability must be proved in all possible scenarios [see sidebar "Firewalls under test"]. Nevertheless, some measure of the degree of a system’s security is necessary.

A first step in this direction is firewall product certification. The International Computer Security Association (ICSA), Carlisle, Pa. (Web site, http://www.icsa.net), has established a laboratory that evaluates and certifies firewall products and systems. The association’s goal is to make the on-line world a safer place and promote its growth by reducing real and perceived risks to computer systems protected by firewalls. The danger in certification, though, is that it can lead to a false sense of security if it simply rubber-stamps products after they have been tested in an artificial laboratory environment. Those labs are generally quite different from real operational environments.

Some organizations offer comparisons of firewall products. For example, the Computer Security Institute, San Francisco, Calif. (http://www.gocsi.com/firewall.htm), periodically publishes a firewall product matrix organized by type and product information, It covers administration and reports, alarms and transparency, authentication and encryption, and proxies, gateways, and servers. While the matrix is neither an evaluation of products nor a comprehensive list of available products (it is primarily generated from marketing brochures), it can serve as a starting point for a review of products.

Another company, Fortified Networks Inc., Carmel, Ind., (http://www.fortified.com/fwcklist.html), offers a firewall evaluation checklist in two versions: a free one and a commercial one that goes into more detail. Both are vendor-neutral; with them, a user can perform a side-by-side comparison of various firewall products.

In addition, various firewall product comparisons have been published by magazines covering security, networking, and PCs. They generally focus on the features and specifications important to the magazine's specialty. For example, *PC Magazine* has evaluated firewall products in terms of their compatibility with Windows NT.

Prospective buyers can also elect to do their own product evaluations by systematically seeking answers to the questions in an evaluation checklist [see sidebar "Firewall product evaluation: a checklist"]. Some questions can be answered simply by consulting suppliers' marketing information, while others may require research and expert knowledge. An evaluator may assign different weights to each of the categories in the checklist to reflect the company's individual priorities.

Once a firewall product has been selected and installed, a system does not automatically become and stay secure. Keeping the firewall up to date is vital. First and foremost, a firewall owner should sign a maintenance agreement with the supplier so that the latest versions of the firewall software arrive promptly; updates can fix both security and performance problems.

If the firewall is hosted on a standard operating system like Windows NT or Sun Solaris, the user should be sure to install the latest security patches released by the operating system vendor.

Finally, the user should stay in touch with others in the security field to keep abreast of new developments. Recommended mailing lists on the Internet are Firewalls, Firewall Wizards, BugTraq, and NT Security. There are also many Usenet news groups devoted to firewalls and security; they can be found by checking the Usenet comp.security hierarchy on the Web. By staying alert, users can prevent problems instead of having to clean up the mess later.

In the near future, several technological trends are bound to affect firewall users. They should be watched closely and adopted as they develop:

- Advanced network technologies such as asynchronous transfer mode (ATM) will require much faster packet-filtering mechanisms because of their higher speed (155 Mb/s versus 1.5 Mb/s).
- Greater heterogeneity in corporate networks and the Internet is likely to lead to unforeseen complexity in filtering rules and the necessity for application-level proxies.
- Firewalls will become more integrated with the networks they protect, as routers, mainframes, and PCs start to incorporate firewalls and firewall characteristics.

Remember, too, that all access points of communication traffic to and from a domain have to be guarded. In an age of laptops, wireless communication, easily available Internet service provider subscriptions, desktop modems, and hand-held computing gadgets, the security perimeter is convoluted indeed. Addressing the main arteries and ignoring the nontraditional ones is like securely bolting the front door of a house but leaving the back door open.

A last point worth remembering is that an Internet firewall does not offer any protection from insider threats. It is like a moat around a medieval castle; it helps to keep an outside enemy outside, but it can do nothing to deter an enemy within.

To probe further

In a ground-breaking book, *Firewalls and Internet Security: Repelling the Wily Hacker* (Addison-Wesley, Reading, Mass., 1994), William R. Cheswick and Steven M. Bellovin discuss the processes, issues, and activities surrounding the implementation of an Internet firewall at AT&T Corp.

The classic introduction to transmission control protocol/Internet protocol (TCP/IP) is Douglas E. Comer's *Internetworking with TCP/IP: Principles, Protocols, and Architecture,* third edition (Prentice-Hall, Englewood Cliffs, N.J., 1995). It starts at the physical layer and progresses through the application layer to Internet security and the Next-Generation Internet protocol.

D. Brent Chapman and Elizabeth D. Zwicky, in *Building Internet Firewalls* (O'Reilly & Associates, Sebastopol, Calif., 1995), discuss network security, building firewalls, and keeping a site secure. They include practical examples and specific configuration information for many firewall situations.

Warwick Ford's *Computer Communications Security: Principles, Standard Protocols, and Techniques* (Prentice-Hall, Englewood Cliffs, N.J., 1993) explains modern standardized methods of achieving network security in both TCP/IP and open systems interconnection environments. Ford gives a technical tutorial introduction to computer network security and describes security standards, protocols, and techniques.

A down-to-earth guide, *Practical UNIX and Internet Security*, second edition, by Simpson Garfinkel and Gene Spafford (O'Reilly and Associates, Sebastopol, Calif., 1996), explains dangers to and methods for keeping systems and data secure. It covers computer security basics, user responsibilities, system security, network security, firewalls, and ways of handling security incidents.

Larry J. Hughes Jr., in his *Actually Useful Internet Security Techniques* (New Riders Publishing, Indianapolis, Ind., 1995), discusses computer security basics and specifics of a wide range of Internet security techniques, including firewalls.

In *Implementing Internet Security* (New Riders Publishing, Indianapolis, Ind., 1995) William Stallings and his coauthors discuss the inconsistencies, weaknesses, and breaches in existing computer security implementations and take a comprehensive look at Internet and network security, including firewalls.

Christoph L. Schuba's dissertation "On the modeling, design, and implementation of firewall technology" (Purdue University, Lafayette, Ind., December 1997) describes a reference model for firewall technology and discusses the life cycle model of firewall design in detail.

Karanjit Siyan and Chris Hare, in their *Internet Firewalls and Network Security* (New Riders Publishing, Indianapolis, Ind.,1996) offer guidance about security and the risks involved in connecting to the Internet, building your own firewall, and developing a solid understanding of concepts, passwords, and standards.

For a comprehensive treatment of the many cryptographic mechanisms applicable to firewalls, see Bruce Schneier's *Applied Cryptography,* second edition (John Wiley & Sons, New York, 1995).

For practical and cost-effective solutions to information security protection, including a firewall product matrix and security education and training, contact the Computer Security Institute (CSI), 600 Harrison St., San Francisco, CA 94107; 415-905-2626; Web, http://www.gocsi.com/.

The International Computer Security Association (ICSA), 1200 Walnut Bottom Rd., Carlisle, PA 17013; 717-258-1816; Web, http://www.icsa.net/, is an independent organization that promotes continuous improvement of commercial digital security through the application of its risk framework, and its continuous certification model, to certification, research, and related activities. The ISCA certifies products, systems, and people.

The Computer Operations, Audit, and Security Technology (Coast) Project at Purdue University focuses on real-world needs and limitations, especially those of existing computing systems. It has close ties to researchers and engineers in major companies and government agencies. For further information, check its Web site at http://www.cs.purdue.edu/coast/.

InterNIC Directory and Database Services (http://ds.internic.net/) maintains several archives that contain documents related to the Internet, including Internet requests for comments (RFCs) and informational bulletins (FYIs) and the Internet Engineering Task Force (IETF) activities. The Internet Documentation Repository Web site is http://ds2.internic.net/ds/dspg0intdoc.html. The various RFCs mentioned in this article can be found at that location.

The National Institute of Standards and Technology (NIST) Computer Security Resource Clearinghouse (http://csrc.nist.gov/) publishes national security standards, such as the Data Encryption Standard (DES), and computer security bulletins, such as "Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls" (NIST Special Publication 800-10, by John P. Wack and Lisa J. Carnahan, 1995).

Check out the Usenet comp.security hierarchy (http://www.liszt.comnews/comp/security) for security-related mailing lists. To subscribe to a mailing list devoted to discussions of the design, construction, operation, maintenance, and philosophy of firewall systems, send an e-mail to majordomo@greatcircle.com. Archives are available at http://www.greatcircle.com/firewalls. To subscribe to a mailing list about firewalls in the academic environment, send e-mail to majordomo@net.tamu.edu. To get onto a mailing list dealing with Unix and firewall holes in great detail, send an e-mail request to bugtraq-request@fc.net.

About the authors

Steve Lodin is an information security services manager in the Indianapolis office of Ernst & Young LLP, providing information security consulting services, including Internet firewall penetration testing. On a General Motors fellowship, he received an MS in computer sciences in 1996 from Purdue University, Indianapolis, Ind., where he was a member of the Coast (Computer Operations, Audit, and Security Technology) Project and discovered the vulnerability of the Kerberos Version 4 random number generator. His research interests include intrusion detection, firewalls, and cryptography. Steven.Lodin@ey.com is his e-mail address.

Christoph Schuba is a staff engineer in the network security research group at Sun Microsystems Inc.’s SunLabs, Palo Alto, Calif. He studied mathematics and management information systems at the

University of Heidelberg and the University of Mannheim, in Germany, where he received the Vordiplom in 1991. As a Fulbright scholar, he earned an MS in 1993 at Purdue University, where he was a member of the Coast research group. He received a Ph.D. from Purdue in 1997, performing part of his dissertation research in the Computer Science Laboratory at the Xerox Palo Alto Research Center. His research interests include network and computer system security and distributed systems. His e-mail address is christoph.schuba@sun.com.

Sidebar: Defining terms

ActiveX: a computer scripting language, developed by Microsoft Corp., Redmond, Wash., by which small programs can be downloaded and executed.

Authenticity: the quality of being genuine both in apparent origin and in content.

Datagram: a data packet traveling through the Internet.

Domain: a set of interconnected networks, gateways, and hosts.

Gateway, router, switch: a device that forwards information from one network to another.

Integrity: the quality of being in the original form, without any modifications such as insertions, replacements, or deletions.

Java: a computer language, developed by Sun Microsystems Inc., Palo Alto Calif., for writing small programs that can be downloaded and executed.

Network: a communication system that allows computers and other electronic devices attached to it to exchange data.

Proxy: a technique for relaying data in which a process that allows easy verification is substituted for the original relaying process.

Security policy: the security requirements defined for a given system--a set of standards, rules, and practices.

State: the condition of a process, protocol, or system at a given time.

Sidebar: Firewalls under test

Management and auditors often require that the security of a firewall installation be ensured, so a variety of test and evaluation procedures is available for the job. A big advantage of firewall testing is that it can be applied periodically to renew confidence in an installation--no small comfort to system administrators.

While each of the testing methods has its own advantages and disadvantages, no single method leads to a perfect firewall. They can, however, be applied selectively to suit the occasion.

Two of the most popular approaches to checking out the actual capabilities of firewalls are generally known as design-oriented testing and operational testing.

In design-oriented testing, configurations and release levels of the firewall's components and accessible network services undergo combined manual and automated testing. This method examines firewalls first at a high level and from there passes to lower levels with increasing detail, as far as it is sensible to go.

This approach to testing yields high-quality functional reviews, but it is time-consuming and expensive, requiring the participation of experts in all the components of the firewall system. Often it is not possible to review a product in detail because it is not distributed as an open platform.

The design-oriented testing method is applied after a security policy has been defined and the firewall is deployed and configured. It consists of five steps:

- Comparison of the actual implementation to the owner's plan.
- Manual investigation of the configuration through management interfaces.
- Operation testing, in which the analyst uses tools to probe the firewall and the network behind the firewall for exposed services, determining whether the filtering rules perform the actions they are supposed to.
- Examination of the allowed services to make sure that all available patches to known vulnerabilities are applied.
- Assignment of a periodic review cycle by the same process.

What this method produces are a policy review; an implementation review of firewall configuration and component release levels; an assessment of services, yielding a list of approved services, their configurations, and their release levels; and a review cycle. (To the extent that they contribute to security, services are part of the firewall system and are subject to review as much as the other parts of the system.)

Whenever mismatches between the implementation and the security policy show up, they are resolved by changing either the implementation or the policy. Be forewarned, however, that revamping the policy can be a difficult political process, as it may involve realigning organizational goals to meet security objectives.

The other method of testing--operational testing--is also known by such names as penetration testing and the "tiger team" approach. No matter what it is called, its application requires that the deployed systems be probed by experts for possible vulnerabilities: the firewalls are given a black-box type of evaluation after they have been installed.

Like software testing, this method can ensure proper behavior of the system in certain common scenarios and situations, and can uncover weaknesses. Still, one can never be certain of having tested enough or of not having missed a major flaw. What is more, operational testing also requires a well-thought-out and well-articulated security policy, which is usually difficult to obtain.

Sidebar: Firewall product evaluation: a checklist

Analyzing commercial firewall products is best done by systematically finding answers to a long series of questions. The process helps a designer consider products objectively and choose those that are best for the security problem at hand. A suggested list of questions follows, grouped according to information category.

Identification: Who are the manufacturer and vendor, what is the product version, and what type of firewall mechanism is it? (This information is readily available from manufacturers' brochures and data sheets.)

Education/documentation: Is the product documentation comprehensive, clear, concise, and well organized? Is it in tutorial or manual style or both? Is product training available? Is training included in the purchase price of the product? Is it provided by the manufacturer or by a consultant? Is technical support available? How qualified is the support, and at what hours is it available? Do the technical support people answer questions and address problems promptly and correctly? Are technical support or service contracts included in the purchase price?

Functionality: How does the product integrate with existing systems? Is it essentially a plug and-play product, or does it require an extensive setup and adjustment to work well with existing systems? What software platforms, such as operating systems, are compatible? Can the product be readily integrated with other firewall services and support tools? What local network topologies are possible, supported, or required: Internet/intranet, demilitarized zones, virtual private networks, network address translation? What physical network topologies are supported: Ethernet, Fast Ethernet, token ring, asynchronous transfer mode? How does the product interact with other firewall products? How transparent is it to users?

Does the product come as an open system? That is, if it is a software-based firewall, is its source code included in its distribution? What are the application programmer interfaces (APIs) and how extensible are they? Does the product support the content vectoring protocol, an API that allows external programs to operate on a firewall? The answer can help determine what add-on products like virus scanners and World Wide Web filters are compatible.

What protocols are covered? Network protocols such as IP, IPX, Appletalk, XNS, SNA, and X.25 may be of interest. What management protocols are supported--SNMP, SNMP-II, Bridge, OOB? What is the base for the management agent: http, Telnet, SNMP, DECnet, or remote terminal?

Reports and audits: What types of reports are available: usage, operation, incident, summary? Are they available in per-user and per-service formats? Can the data be exported to external databases? Are the reporting mechanisms of the firewall flexible, extensible, and configurable in detail? Is real-time notification possible--by e-mail or paging, for example? What audit media are supported--hard copy, write-once/read-multiple (WORM) drives, remote logging? (WORM technology ensures that audit data, once generated, cannot be erased by intruders to cover their tracks.)

Are audit analysis tools for the reports available or included? Is software for generating and individualizing reports available or included? If intrusion detection is part of the system, how well does it work--what is the number of false positives?

Attacks and responses: What network-based attack scenarios does the product protect against? (Attacks are often based on address spoofing, sequence number prediction, session hijacking, fragmentation, source routing, spoofed naming-service (such as DNS) packets, spoofed routing packets, spoofed control packets, port scanning, "Christmas tree" packets, and/or spoofed multicast and broadcast packets. Usually it is necessary to perform penetration testing--that is, subject the product to an attack--to see how it will behave. Does the product offer counterattack or counterintelligence capability, such as information gathering about the apparent origin sites of malicious packets?

What is the fault tolerance of the product? How does it behave under heavy loads and congestion, after a power failure, and during boot time? (It is known that some firewall products malfunction under some of these conditions.) Does the product contain automated integrity checks, and perhaps congestion control mechanisms?

Does the product recognize data content? Does it determine whether viruses, executable code, Java script or ActiveX code, or mail attachments have been transmitted?

Does the product provide encryption? Which encryption algorithms are available and what are their key lengths? Does it support firewall-to-firewall encryption? Encryption of administrative dial-up connections? User-to-firewall encryption? What is the key-exchange protocol and the frequency of key exchange? How easy is it to exchange keys. Is it compliant with the so-called IPsec protocols developed by the IP Security Working Group of the Internet Engineering Task Force (IETF) ?

Administrative concerns: How secure and flexible is administrative access to the firewall product? Does it support authentication mechanisms like Bellcore S/Key, Security Dynamics SecurID, Digital Pathways SecureNet Key, CryptoCard RB-1, or Enigma Logic SafeWord? Can security functions be accessed by a dial-up connection? Is there a text-only administrative interface (one that, for security reasons, is not X-Windows-based)? Can the administration separate management tasks and delegate roles?

If your organization has locations outside the country in which it is based, can the firewall product be exported there?

How does the product appear to the external network? Is it network-addressable, or are there no mechanisms for accessing it and attacking it over the network?

How well does it interact with other firewalls? Can loads and bandwidths be balanced among the firewalls? How well does the administrative interface work with multiple firewalls?

What is the product’s bandwidth or aggregate throughput, as measured by its packet-forwarding rate? What performance benchmarks are available from independent testing laboratories? How many firewalls are needed to handle saturated T1, T3, or Ethernet traffic? What delay does the product introduce by encryption?

What filters does the product offer? Are input and output filters separated? Are there filters for protocols, addresses, services, and user-defined patterns? What is the filtering rate in packets per second? Is it easy to specify and implement a filtering policy?

If the product is an application-level gateway, does it offer per-service features for access control, authentication, logging, and auditing? How easy is it to define and implement a user-definable or generic proxy service?

Implementation and afterward: What are the installation requirements? What software is prerequisite? Is third-party code required? What hardware is prerequisite--routers, hosts, electric power, specialized network interfaces? Will any existing routers or hosts have to be replaced or augmented? What administrative infrastructure is needed?

How easy are the hardware and software to install? Do the default settings make sense and are they secure? Are services enabled or disabled by default? Is logging enabled or disabled by default?

Is there a defined upgrade schedule for the product? Does the vendor provide quick fixes for security issues? What is the upgrade distribution mechanism--tapes, diskettes, on-line?

The bottom line: Last but certainly not least, what is the price tag for the hardware, software, extra equipment, installation and migration, training (basic and advanced), service contracts, and ongoing administration? What benefits does the product's warranty provide?

Spectrum editor: Richard Comerford

IEEE Spectrum February 1998 Volume 35 Number 2