COMPARING FIREWALL TECHNOLOGIES

Per Thorsheim

INSIDE

Firewall Technologies Explained; Network Level Firewalls: Packet Filters; Stateful Inspection Firewalls; Application-Level Firewalls; What the Market Wants versus What the Market Really Needs; Perimeter Defense and How Firewalls Fit In; Practical Example of Missing Egress (Outbound) Filtering; Common Mistakes that Lead to System and Network Compromises; Intrusion-Detection Systems and Firewalls

In early January 2001, a new web page was launched. It was named Netscan,[1] and the creators had done quite a bit of work prior to launching their Web site. Actually, the work was quite simple, but time-consuming. They had pinged the entire routed IPv4 address space; or to be more exact, they pinged every IP address ending with .0 or .255. For each PING sent, they expected one PING REPLY in return. And for each network that replied with more than one packet, they counted the number of replies and put the data into a database. All networks that did reply with more than one packet for each packet sent were considered to be an amplifier network. After pinging the entire Internet (more or less), they published on their Web site a list of the 1024 worst networks, including the e–mail address for the person responsible for the IP address and its associated network. The worst networks were those networks that gave them the highest number of replies to a single PING, or the best amplification effect.

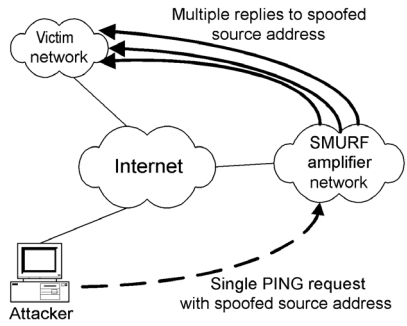
The security problem here is that it is rather easy to send a PING request to a network, using a spoofed source IP address. And when the recipient network replies, all those replies will be sent to the source address as given in the initial PING. As shown in Exhibit 1, the attacker can

PAYOFF IDEA

Firewalls and firewall technologies by themselves cannot be trusted, at least not in our present Internet age of communications with hackers hiding in every corner. Hackers tunneling data through allowed protocols and ports using encryption schemes to hide their tracks, can easily bypass today's firewalls. Security professionals should understand that a firewall, as part of a consistent overall security architecture, is still an important part of the network security in a company.

Auerbach Publications

EXHIBIT 1 — Attacker Using Spoofed PING Packets to Flood a Network Using a Vulnerable Intermediary Network



flood the Internet connection of the final recipient by repeating this procedure continuously.

In fact, the attacker can use an ISDN connection to create enough traffic to jam a T3 (45–Mbit) connection, using several SMURF amplifier networks to launch the attack. And as long as there are networks that allow such amplification, a network can be the target of the attack even if the network does not have the amplification problem itself, and there is not much security systems such as firewalls can do to prevent the attack.

This type of attack has been used over and over again to attack some of the biggest sites on the Internet, including the February 2000 attacks against Yahoo, CNN, Ebay, and Amazon.

Today, there are several Web sites that search for SMURF amplifier networks and publish their results publicly. In a presentation given in March 2001, this author pointed out the fact that the number of networks not protected from being used as such amplifiers had increased more than 1000 percent since January 2001.

One of the interesting findings from these attacks was that routers got blamed for the problems — not firewalls. And they were correct; badly configured Internet routers were a major part of the problem in these cases. Even worse is the fact that the only requirement for blocking this specific PING-based attack was to set one parameter in all routers connecting networks to the Internet. This has now become the recommended default in RFC 2644/BCP 34, "Changing the Default for Directed Broadcast in Routers." Security professionals should also read RFC 2827/BCP 0038, "Network Ingress Filtering: Defeating Denial-of-Service Attacks Which Employ IP Source Address Spoofing," to further understand spoofing attacks.

Another interesting observation after these attacks was President Clinton's announcement of a National Plan for Information Systems Protection, with valuable help from some of the top security experts in the

United States. In this author's opinion, this serves as the perfect example of who should be at the top and responsible for security — the board of directors and the CEO of a company.

Finally, Web sites such as CNN, Yahoo, and Amazon all had firewalls in place, yet that did not prevent these attacks. Thus, a discussion of firewall technologies and what kind of security they can actually provide is in order.

FIREWALL TECHNOLOGIES EXPLAINED

The Internet Firewalls FAQ[2] defines two basic types of firewalls: network-layer firewalls and application-layer firewalls (also referred to as application proxy firewalls, or just proxies). For this chapter, stateful inspection firewalls are defined as a mix of the first two firewall types, in order to make it easier to understand the similarities and differences between them.

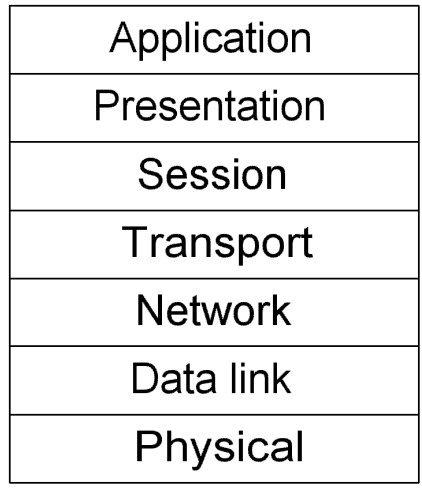
The reader may already be familiar with the OSI layer model, in which the network layer is layer 3 and the application layer is at layer 7, as shown in Exhibit 2.

A firewall can simply be illustrated as a router that transmits packets back and forth between two or more networks, with some kind of security filtering applied on top.

Network-Level Firewalls: Packet Filters

Packet filter firewalls are very often just a router with access lists. In its most basic form, a packet filter firewall controls traffic based on the source and destination IP address of each IP packet and the destination

EXHIBIT 2 — The OSI Seven-Layer Model



port. Many packet filter firewalls also allow checking the packets based on the incoming interface (is it coming from the Internet, or the internal network?). They may also allow control of the IP packet based on the source port, day and time, protocol type (TCP, UDP, or ICMP), and other IP options as well, depending on the product.

The first thing to remember about packet filter firewalls is that they inspect every IP packet by itself; they do not see IP packets as part of a session. The second thing to remember about packet filter firewalls is that many of them, by default, have a fail-open configuration, meaning that, by default, they will let packets through unless specifically instructed not to. And finally, packet filters only check the HEADER of a packet, and not the DATA part of the packet. This means that techniques such as tunneling a service within another service will easily bypass a packet filter (e.g., running Telnet on port 80 through a firewall where the standard Telnet port 23 is blocked, but HTTP port 80 is open. Because the packet filter only sees source/destination and port number, it will allow it to pass).

Why Use Packet Filter Firewalls? Some security managers may not be aware of it, but most probably there are lots of devices already in their network that can do packet filtering. The best examples are various routers. Most (if not all) routers today can be equipped with access lists, controlling IP traffic flowing through the router with various degrees of security. In many networks, it will just be a matter of properly configuring them for the purpose of acting as a packet filter firewall. In fact, the author usually recommends that all routers be equipped with at least a minimum of access lists, in order to maintain security for the router itself and its surroundings at a minimal level. Using packet filtering usually has little or no impact on throughput, which is another plus over the other technologies. Finally, packet filter firewalls support most (if not all) TCP/IP-based services.

Why Not Use Packet Filter Firewalls? Well, they only work at OSI layer 3, or the network layer as it is usually called. Packet filter firewalls only check single IP packets; they do not care whether or not the packet is part of a session. Furthermore, they do not do any checking of the actual contents of the packet, as long as the basic header information is okay (such as source and destination IP address). It can be frustrating and difficult to create rules for packet filter firewalls, and maintaining consistent rules among many different packet filter firewalls is usually considered very difficult. As previously mentioned, the typical fail-open defaults should be considered dangerous in most cases.

Stateful Inspection Firewalls

Basically, stateful inspection firewalls are the same thing as packet filter firewalls, but with the ability to keep track of the state of connections in addition to the packet filter abilities. By dynamically keeping track of whether a session is being initiated, currently transmitting data (in either direction), or being closed, the firewall can apply stronger security to the transmission of data. In addition, stateful inspection firewalls have various ways of handling popular services such as HTTP, FTP, and SMTP. These last options (which there are many variants of from product to product) enable the firewall to actually check whether or not it is HTTP traffic going to TCP port 80 on a host in a network by "analyzing" the traffic. A packet filter will only assume that it is HTTP traffic because it is going to TCP port 80 on a host system; it has no way of actually checking the DATA part of the packet, while stateful inspection can partially do this.

A stateful inspection firewall is capable of understanding the opening, communication, and closing of sessions. Stateful inspection firewalls usually have a fail-close default configuration, meaning that they will not allow a packet to pass if it does not know how to handle the packet. In addition to this, they can also provide an extra level of security by "understanding" the actual contents (the data itself) within packets and sessions, compared to packet filters. This last part only applies to specific services, which may be different from product to product.

Why Use Stateful Inspection Firewalls? Stateful inspection firewalls give high performance and provide more security features than packet filtering. Such features can provide extra control of common and popular services. Stateful inspection firewalls support most (if not all) services transparently, just like packet filters, and there is no need to modify client configurations or add any extra software for them to work.

Why Not Use Stateful Inspection Firewalls? Stateful inspection firewalls may not provide the same level of security as application-level firewalls. They let the server and the client talk "directly" to each other, just like packet filters. This may be a security risk if the firewall does not know how to interpret the DATA contents of the packets flowing through the firewall. Even more disturbing is the fact that many people consider stateful inspection firewalls to be easier to configure wrongly, compared to application-level firewalls. This is due to the fact that packet filters and stateful inspection firewalls support most, if not all, services transparently, while application-level firewalls usually support only a very limited number of services and require modification to client software in order to work with non-supported services.

In a white paper from Network Associates,[3] the Computer Security Institute (CSI) was quoted as saying, "It is quite possible, in fact trivial,

to configure stateful inspection firewalls to permit dangerous services through the firewall…. Application proxy firewalls, by design, make it far more difficult to make mistakes during configuration."

Of course, it should be unnecessary to say that no system is secure if it is not configured correctly.

And human faults and errors are the number one, two, and three reasons for security problems, right?

Application-Level Firewalls

Application-level firewalls (or just proxies) work as a "man-in-the-middle," where the client asks the proxy to perform a task on behalf of the client. This could include tasks such as fetching Web pages, sending mail, retrieving files using FTP, etc. Proxies are application specific, meaning that they need to support the specific application (or, more exactly, the application-level protocol) that will be used. There are also standards for generic proxy functionality, with the most popular being SOCKS. SOCKS was originally authored by David Koblas and further developed by NEC. Applications that support SOCKS will be able to communicate through firewalls that also support the SOCKS standard.[4]

Similar to a stateful inspection firewall, the usual default of an application-level firewall is fail-close, meaning that it will block packets/sessions that it does not understand how to handle.

Why Use Application-Level Firewalls? First of all, they provide a high level of security, primarily based on the simple fact that they only support a very limited number of services; however, they do support most, if not all, of the usual services that are needed on a day-to-day basis. They understand the protocols at the application layer and, as such, they may block parts of a protocol (allow receiving files using FTP, but denying sending files using FTP as an example). They can also detect and block vulnerabilities, depending on the firewall vendor and version.

Furthermore, there is no direct contact being made between the client and the server; the firewall will handle all requests and responses for the client and the server. With a proxy server, it is also easy to perform user authentication, and many security practitioners will appreciate the extensive level of logging available in application-level firewalls.

For performance reasons, many application-level firewalls can also cache data, providing faster response times and higher throughput for access to commonly accessed Web pages, for example. The author usually does not recommend that a firewall do this because a firewall should handle the inspection of traffic and provide a high level of security. Instead, security practitioners should consider using a stand-alone caching proxy server for increasing performance while accessing common Web sites. Such a stand-alone caching proxy server may, of course,

also be equipped with additional content security, thus controlling access to Web sites based on content and other issues.

Why Not Use Application-Level Firewalls? By design, application-level firewalls only support a limited number of services. If support for other applications/services/protocols is desired, applications may have to be changed in order to work through an application-level firewall. Given the high level of security such a firewall may provide (depending on its configuration, of course), it may have a very negative impact on performance compared to packet filtering and stateful inspection firewalls.

What the Market Wants versus What the Market Really Needs

Many firewalls today seem to mix these technologies together into a simple and easy-to-use product. Firewalls try to be a "turnkey" or "all-in-one" solution. Security in a firewall that can be configured by more or less plugging it in and turning it on is something in which the author has little faith. And, the all-in-one solution that integrates VPN, antivirus, content security/filtering, traffic shaping, and similar functionality is also something in which the author has little trust. In fact, firewalls seem to get increasingly complex in order to make them easier to configure, use, and understand for the end users. This seems a little bit wrong; by increasing the amount of code in a product, the chances of security vulnerabilities in the product increase, and most probably exponentially.

In the author's opinion, a firewall is a "black box" in a network, which most regular users will not see or notice. Users should not even know that it is there.

The market decides what it wants, and the vendors provide exactly that. But does the market always know what is good for it? This is a problem that security professionals should always give priority to — teaching security understanding and security awareness.

Firewall Technologies: Quick Summary

As a rule of thumb, packet filters provide the lowest level of security, but the highest throughput. They have limited security options and features and can be difficult to administrate, especially if there is a large number of them in a network.

Stateful inspection firewalls provide a higher level of security, but may not give the same throughput as packet filters. The leading firewalls in the market today are stateful inspection firewalls, often considered as the best mix of security, manageability, throughput, and transparent integration into most environments.

Application-level firewalls are considered by many to give the highest level of security, but will usually give less throughput compared to the two other firewall technologies.

In any case, security professionals should never trust a firewall by itself to provide good security. And no matter what firewall a company deploys, it will not provide much security if it is not configured correctly. And that usually requires quite a bit of work.

PERIMETER DEFENSE AND HOW FIREWALLS FIT IN

Many people seem to believe that all the bad hackers are "out there" on the Internet, while none of their colleagues in a firm would ever even think of doing anything illegal, internally or externally. Sadly, however, there are statistics showing that internal employees carry out maybe 50 percent of all computer-related crime.

This is why it is necessary to explain that security in a firewall and its surrounding environment works two ways. Hackers on the Internet are not allowed access to the internal network, and people (or hostile code such as viruses and Trojans) on the internal network should be prevented from sending sensitive data to the external network. The former is much easier to configure than the latter. As a practical example of this, here is what happened during an Internet penetration test performed by the author some time ago.

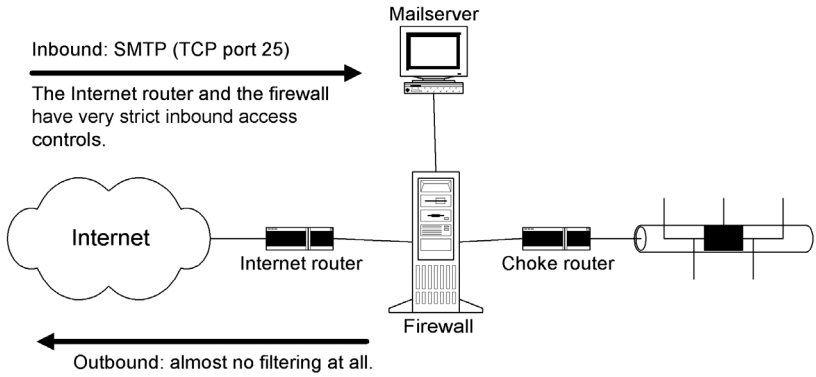
Practical Example of Missing Egress (Outbound) Filtering

The client was an industrial client with a rather simple firewall environment connecting them to the Internet. They wanted a high level of security and had used external resources to help configure their Internet router act as a packet filter firewall, in addition to a stateful inspection firewall on the inside of the Internet router, with a connection to the internal network. They had configured their router and firewall to only allow e–mail (SMTP, TCP port 25) back and forth between the Internet and their antivirus (AV) e–mail gateway placed in a demilitarized zone (DMZ) on the stateful inspection firewall. The antivirus e–mail gateway would check all in- and outgoing e–mail before sending it to the final recipient, be it on the internal network or on the Internet. The router was incredibly well configured; inbound access lists were extremely strict, only allowing inbound SMTP to TCP port 25. The same thing was the case for the stateful inspection firewall.

While testing the antivirus e–mail gateway for SMTP vulnerabilities, the author suddenly noticed that each time he connected to the SMTP connector of the antivirus e–mail gateway, it also sent a Windows NetBIOS request in return, in addition to the SMTP login banner.

This simple fact reveals a lot of information to an unauthorized person (see Exhibit 3). First of all, there is an obvious lack of egress

EXHIBIT 3 — Missing Egress Filtering in the Router and the Firewall May Disclose Useful Information to Unauthorized People



(outbound) filtering in both the Internet router and the firewall. This tells us that internal systems (at least this one in the DMZ) can probably do NetBIOS communication over TCP/IP with external systems. This is highly dangerous for many reasons. Second, the antivirus e–mail gateway in the DMZ is installed with NetBIOS, which may indicate that recommended good practices have not been followed for installing a Windows server in a high-security environment. Third, it may be possible to use this system to access other systems in the DMZ or on other networks (including the internal network) because NetBIOS is being used for communication among windows computers in a workgroup or domain. At least, this is the author's usual experience when doing Internet penetration testing. Of course, an unauthorized person must break into the server in the DMZ first, but that also proves to be easier than most people want to believe.

How Can One Prevent Such Information Leakage? Security managers should check that all firewalls and routers connecting them to external networks have been properly configured to block services that are considered "dangerous," as well as all services that are never supposed to be used against hosts on external networks, especially the Internet.

As a general rule, security managers should never allow servers and systems that are not being used at the local console to access the Internet in any way whatsoever. This will greatly enhance security, in the way that hostile code such as viruses and Trojans will not be able to directly establish contact with and turn over control of the system to unauthorized persons on any external network.

This also applies to systems placed in a firewall DMZ, where there are systems that can be accessed by external people, even without any kind of user authentication. The important thing to remember here is: who makes the initial request to connect to a system?

If it is an external system making a connection to a mail server in a DMZ on TCP port 25 (SMTP), it is okay, because it is (probably) incoming e–mail. If the mail server in the DMZ makes a connection to an external system on TCP port 25, that is also okay because it does this to send outgoing e–mail. However, if the only purpose of the mail server is to send and receive mail to and from the Internet, the firewalls and even the routers should be configured in accordance with this.

For the sake of easy administration, many people choose to update their servers directly from the Internet; some even have a tendency to sit directly on production servers and surf the World Wide Web without any restrictions or boundaries whatsoever. This poses a high security risk for the server, and also the rest of the surrounding environment, given the fact that (1) Trojans may get into the system, and (2) servers tend to have the same usernames and passwords even if they do not have anything in common except for being in the same physical/logical network.

To quote Anthony C. Zboralski Gaius[5] and his article "Things to do in Cisco land when you're dead" in *Phrack Magazine*[6]:

> It's been a long time since I stopped believing in security. The core of the security problem is really because we are trusting trust (read Ken Thomson's article, Reflections on Trusting Trust). If I did believe in security then I wouldn't be selling penetration tests.

It can never be said that there is a logical link between high security and easy administration, nor will there ever be. Security is difficult, and it will always be difficult.

Common Mistakes that Lead to System and Network Compromises

Many security professionals say that "networks are hard on the outside, and soft on the inside," a phrase this author fully agrees with. The listing that follows shows some of the common weaknesses encountered over and over again.

- Remote access servers (RAS) are connected to the internal network, allowing intruders access to the network just like internal users, as soon as they have a username and password.
- Access lists and other security measures are not implemented in WAN routers and networks. Because small regional offices usually have a lower level of physical security, it may be easier to get access to the office, representing a serious risk to the entire network.
- Many services have default installations, making them vulnerable. They have known weaknesses, such as standard paths of

installation, file and directory permissions are often default everyone full control, etc.

- Employees do not follow written password policies, and password policies are usually written with users (real people) in mind, and not generic system accounts.
- Many unnecessary services are running on various systems without being used. Many of these services can easily be used for denial-of-service (DoS) attacks against the system and across the network.
- Service applications run with administrator privileges, and their passwords are rarely changed from the default value. As an example, there are backup programs in which the program's username and password are the same as the name of the program, and the account has administrative privileges by default. Take a look at some of the default usernames/passwords lists that exist on the Internet; they list hundreds of default usernames and passwords for many, many different systems.[7]
- Companies have trust in authentication mechanisms and use them as their only defense against unauthorized people trying to get access to the various systems in the network. Many companies and people do not seem to understand that hackers do not need a username or password to get access to different systems; there are many vulnerabilities that give them full control within seconds.

Most, if not all, security professionals will recognize many of these as problems that will never go away. At the same time, it is very important to understand these problems, professionals should work continuously to reduce or remove these problems.

When performing penetration testing, common questions and comments include: "How are you going to break into our firewall?" and "You are not allowed to do this and this and that." First of all, penetration testing does not involve breaking into firewalls, just trying to bypass them. Breaking into a firewall by itself may show good technical skills, but it does not really do much harm to the company that owns it. Second, hackers do not have to follow any rules, either given by the company they attack or the laws of the country (or the laws of the many countries they are passing through in order to do the attack over the Internet, which opens up lots more problems for tracking down and punishing the hackers, a problem that many security professionals are trying to deal with already).

What about Security at the Management Workstations? Many companies are deploying extremely tight security into their Internet connection environment and their internal servers. What many of them do wrong is that they forget to secure the workstations that are being used to administrate those highly secured systems. During a recent security

audit of an Internet bank, the author was given an impressive presentation with firewalls, intrusion detection systems, proxies, and lots of other stuff thrown in. When checking a bit deeper, it was discovered that all the high-security systems were managed from specific workstations located on their internal network. All those workstations ("owned" by network administrators) were running various operating systems (network administrators tend to do this…) with more or less default configurations, including default usernames and passwords, SNMP,[8] and various services. All those workstations were in a network mixed with normal users; there were no access restrictions deployed except username/password to get access to those management stations. They even used a naming convention for their internal computers that immediately revealed which ones were being used for "critical system administration." By breaking into those workstations first (Trojans, physical access, other methods), it did not take long to get access to the critical systems.

Intrusion Detection Systems and Firewalls

Lately, more and more companies have been deploying intrusion detection systems (IDS) in their networks. Here is another area in which it is easy to make mistakes. First of all, an IDS does not really help a company improve its security against hackers. An IDS will help a company to better detect and document an attack, but in most cases it will not be able to stop the attack. It is tempting to say that an IDS is just a new term for extensive logging and automated/manual analysis, which have been around for quite some time now.

Some time ago, someone came up with the bright idea of creating IDS that could automatically block various attacks, or reconfigure other systems like firewalls to block the attacks. By doing a spoofing attack (very easy these days), hackers could create a false attack that originated from a trusted source (third party), making the IDS block all communications between the company and the trusted source. And suddenly everybody understood that the idea of such automated systems was probably a bad idea.

Some IDS are signature based, while others are anomaly based. Some IDS have both options, and maybe host and network based agents as well. And, of course, there are central consoles for logging and administrating the IDS agents deployed in the network. (How good is the security at those central consoles?)

- *Problem 1.* Signature-based detection more or less depends on specific data patterns to detect an attack. Circumventing this is becoming easier every day as hackers learn how to circumvent the patterns known by the IDS, while still making patterns that work against the target systems.

- *Problem 2.* Most IDS do not understand how the receiving system reacts to the data sent to it, meaning that the IDS can see an attack, but it does not know whether or not the attack was successful. So, how should the IDS classify the attack and assess the probability of the attack being successful?
- *Problem 3.* IDS tend to create incredible amounts of false alerts, so who will check them all to see if they are legitimate or not? Some companies receive so many alerts that they just "tune" the system so that it does not create that many alerts. Sometimes this means that they do not check properly to see if there is something misconfigured in their network, but instead just turn off some of the detection signatures, thus crippling the IDS of its functions.
- *Problem 4.* Anomaly-based detection relies on a pattern of "normal" traffic and then generates alerts based on unusual activity that does not match the "normal" pattern. What is a "normal" pattern? The author has seen IDS deployments in which the IDS were placed into a network that was configured with all sorts of protocols, unnecessary services and clear-text authentication flying over the wire. The "normal" template became a template for which almost everything was allowed, more or less disabling the anomaly detection capability of the IDS. (This is also very typical for "personal firewalls," which people are installing on their home systems these days.)

IDS can be a very effective addition to a firewall because the IDS is usually better at logging the contents of the attack compared to a firewall, which only logs information such as source/destination, date/time, and other information from the various IP/TCP/UDP headers. Using IDS, it is also easier to create statistics over longer periods of time of hacker activity compared to just having a firewall and its logs. Such statistics may also aid in showing management what the reality is when it comes to hacking attempts and illegal access against the company's systems, as well as raising general security awareness among its users.

On the other hand, an IDS requires even more human attention than a firewall, and a company should have very clearly defined goals with such a system before buying and deploying it. Just for keeping hackers out of your network is not a good enough reason.

GENERAL RECOMMENDATIONS AND CONCLUSIONS

A firewall should be configured to protect itself, in addition to the various networks and systems that it moves data to and from. In fact, a firewall should also "protect" the Internet, meaning that it should prevent internal "hackers" from attacking other parties connected to the Internet, wherever and whoever they are. Surrounding network equipment such

as routers, switches, and servers should also be configured to protect the firewall environment in addition to the system itself.

Security professionals should consider using user authentication before allowing access to the Internet. This will, in many situations, block viruses and Trojans from establishing contact with hosts on the Internet using protocols such as HTTP, FTP, and Telnet, for example.

It may be unnecessary to say, but personal use of the Internet from a company network should, in general, be forbidden. Of course, the level of control here can be discussed, but the point is to prevent users from downloading dangerous content (viruses, Trojans) and sending out files from the internal network using protocols such as POP3, SMTP, FTP, HTTP, and other protocols that allow sending files in ASCII or binary formats.

Finally, other tools should be deployed as well to bring the security to a level that actually matches the level required (or wanted) in the company security policy. In the author's experience, probably less than 50 percent of all firewall installations are doing extensive logging, and less than 5 percent of the firewall owners are actually doing anything that even resembles useful log analysis, reporting, and statistics. To some, it seems like the attitude is "we've got a firewall, so we're safe." Such an attitude is both stupid and wrong.

Firewalls and firewall technologies by themselves cannot be trusted, at least not in our present Internet age of communications with hackers hiding in every corner. Hackers tunneling data through allowed protocols and ports can easily bypass today's firewalls, using encryption schemes to hide their tracks. Security professionals should, nonetheless, understand that a firewall, as part of a consistent overall security architecture, is still an important part of the network security in a company.

The best security tool available is still the human brain. Use it wisely and security will improve.

Notes

1. www.netscan.org.
2. http://www.interhack.net/pubs/fwfaq/, Copyright © Marcus J. Ranum and Matt Curtin.
3. Network Associates, "Adaptive Proxy Firewalls — The Next Generation Firewall Architecture."
4. Note that there are two major versions of SOCKS: SOCKS V4 AND SOCKS V5. Version 4 does not support authentication or UDP proxying, while version 5 does.
5. www.hert.org, quoted with permission.
6. www.phrack.com.
7. http://packetstorm.securify.com/ is a good place to search for such lists, and much more useful information as well.
8. Simple Network Management Protocol, one of the author's favorite ways of mapping large networks fast and easy. Also mentioned as number 10 on the SANS' Institute "Top Ten Vulnerabilities" list at http://www.sans.org/topten.htm.

Per Thorsheim is a senior consultant with PriceWaterhouseCoopers in Bergen, Norway.