Internet Firewalls:

Design and Implementation Issues

1- What are some of the basic design decisions in a firewall?

There are a number of basic design issues that should be addressed by the lucky person who has been tasked with the responsibility of designing, specifying, and implementing or overseeing the installation of a firewall.

The first and most important decision reflects the policy of how your company or organization wants to operate the system: is the firewall in place explicitly to deny all services except those critical to the mission of connecting to the Net, or is the firewall in place to provide a metered and audited method of ``queuing'' access in a non-threatening manner? There are degrees of paranoia between these positions; the final stance of your firewall might be more the result of a political than an engineering decision.

The second is: what level of monitoring, redundancy, and control do you want? Having established the acceptable risk level (i.e., how paranoid you are) by resolving the first issue, you can form a checklist of what should be monitored, permitted, and denied. In other words, you start by figuring out your overall objectives, and then combine a needs analysis with a risk assessment, and sort the almost always conflicting requirements out into a laundry list that specifies what you plan to implement.

The third issue is financial. We can't address this one here in anything but vague terms, but it's important to try to quantify any proposed solutions in terms of how much it will cost either to buy or to implement. For example, a complete firewall product may cost between $100,000 at the high end, and free at the low end. The free option, of doing some fancy configuring on a Cisco or similar router will cost nothing but staff time and a few cups of coffee. Implementing a high end firewall from scratch might cost several man-months, which may equate to $30,000 worth of staff salary and benefits. The systems management overhead is also a consideration. Building a home-brew is fine, but it's important to build it so that it doesn't require constant (and expensive) attention. It's important, in other words, to evaluate firewalls not only in terms of what they cost now, but continuing costs such as support.

On the technical side, there are a couple of decisions to make, based on the fact that for all practical purposes what we are talking about is a static traffic routing service placed between the network service provider's router and your internal network. The traffic routing service may be implemented at an IP level via something like screening rules in a router, or at an application level via proxy gateways and services.

The decision to make is whether to place an exposed stripped-down machine on the outside network to run proxy services for telnet, FTP,

news, etc., or whether to set up a screening router as a filter, permitting communication with one or more internal machines. There are benefits and drawbacks to both approaches, with the proxy machine providing a greater level of audit and, potentially, security in return for increased cost in configuration and a decrease in the level of service that may be provided (since a proxy needs to be developed for each desired service). The old trade-off between ease-of-use and security comes back to haunt us with a vengeance.

2- What are the basic types of firewalls?

Conceptually, there are three types of firewalls:
Network layer
Application layer
Hybrids
They are not as different as you might think, and latest technologies are blurring the distinction to the point where it's no longer clear if either one is ``better'' or ``worse.'' As always, you need to be careful to pick the type that meets your needs.
Which is which depends on what mechanisms the firewall uses to pass traffic from one security zone to another? The International Standards Organization (ISO) Open Systems Interconnect (OSI) model for networking defines seven layers, where each layer provides services that ``higher-level'' layers depend on. In order from the bottom, these layers are physical, data link, network, transport, session, presentation, application.
The important thing to recognize is that the lower-level the forwarding mechanism, the less examination the firewall can perform. Generally speaking, lower-level firewalls are faster, but are easier to fool into doing the wrong thing.
These days, most firewalls fall into the ``hybrid'' category, which do network filtering as well as some amount of application inspection. The amount changes depending on the vendor, product, protocol and version, so some level of digging and/or testing is often necessary.

2.1- Network layer firewalls

These generally make their decisions based on the source, destination addresses and ports (see Appendix 6 for a more detailed discussion of ports) in individual IP packets. A simple router is the ``traditional'' network layer firewall, since it is not able to make particularly sophisticated decisions about what a packet is actually talking to or where it actually came from. Modern network layer firewalls have become increasingly sophisticated, and now maintain internal information about the state of connections passing through them, the contents of some of the data streams, and so on. One thing that's an important distinction about

many network layer firewalls is that they route traffic directly though them, so to use one you either need to have a validly assigned IP address block or to use a ``private internet'' address block [5]. Network layer firewalls tend to be very fast and tend to be very transparent to users.

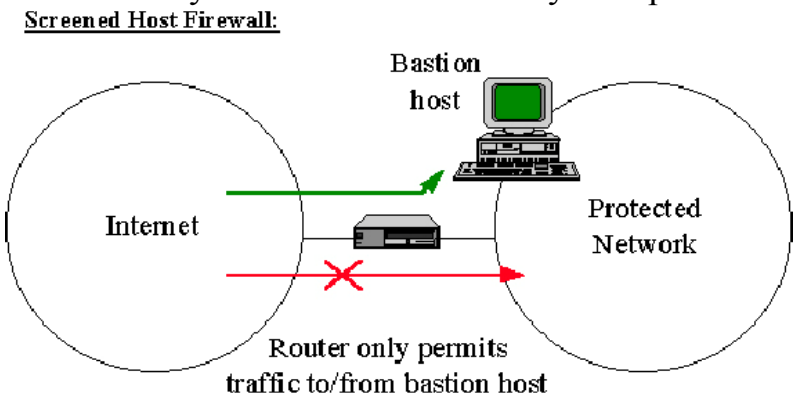

Figure 1: Screened Host Firewall

In Figure 1, a network layer firewall called a ``screened host firewall'' is represented. In a screened host firewall, access to and from a single host is controlled by means of a router operating at a network layer. The single host is a bastion host; a highly-defended and secured strong-point that (hopefully) can resist attack.

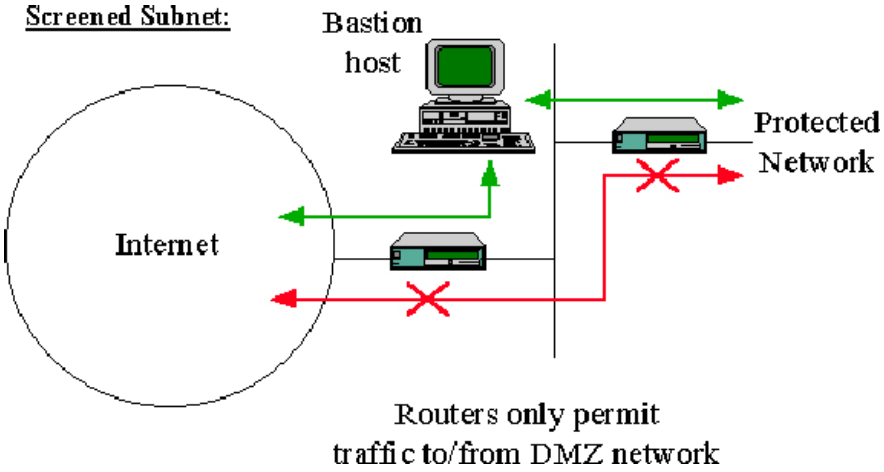

Figure 2: Screened Subnet Firewall

Example Network layer firewall: In Figure 2, a network layer firewall called a ``screened subnet firewall'' is represented. In a screened subnet firewall, access to and from a whole network is controlled by means of a router operating at a network layer. It is similar to a screened host, except that it is, effectively, a network of screened hosts.

2.2- Application layer firewalls

These generally are hosts running proxy servers, which permit no traffic directly between networks, and which perform elaborate logging and auditing of traffic passing through them. Since the proxy applications are software components running on the firewall, it is a good place to do lots

of logging and access control. Application layer firewalls can be used as network address translators, since traffic goes in one ``side'' and out the other, after having passed through an application that effectively masks the origin of the initiating connection. Having an application in the way in some cases may impact performance and may make the firewall less transparent. Early application layer firewalls such as those built using the TIS firewall toolkit, are not particularly transparent to end users and may require some training. Modern application layer firewalls are often fully transparent. Application layer firewalls tend to provide more detailed audit reports and tend to enforce more conservative security models than network layer firewalls.

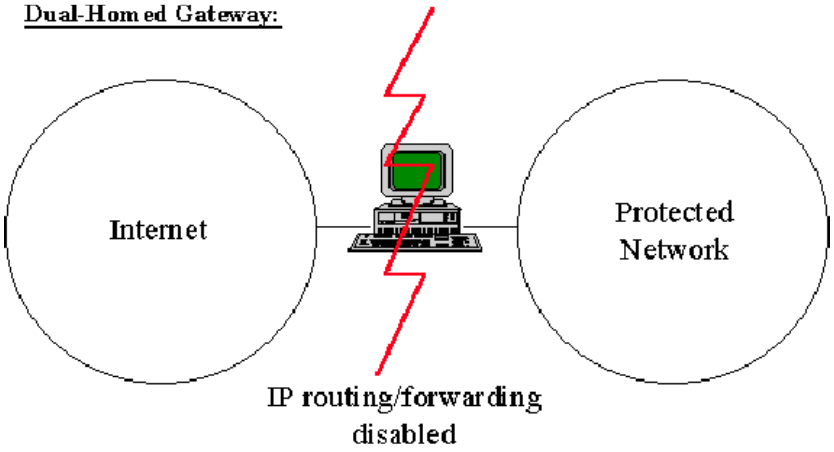

Figure 3: Dual Homed Gateway

Example Application layer firewall: In Figure 3, an application layer firewall called a ``dual homed gateway'' is represented. A dual homed gateway is a highly secured host that runs proxy software. It has two network interfaces, one on each network, and blocks all traffic passing through it.

Most firewalls now lie someplace between network layer firewalls and application layer firewalls. As expected, network layer firewalls have become increasingly ``aware'' of the information going through them, and application layer firewalls have become increasingly ``low level'' and transparent. The end result is that now there are fast packet-screening systems that log and audit data as they pass through the system. Increasingly, firewalls (network and application layer) incorporate encryption so that they may protect traffic passing between them over the Internet. Firewalls with end-to-end encryption can be used by organizations with multiple points of Internet connectivity to use the Internet as a ``private backbone'' without worrying about their data or passwords being sniffed. (IPSEC, described in Section 2.6, is playing an increasingly significant role in the construction of such virtual private networks.)

3- What are proxy servers and how do they work?

A proxy server (sometimes referred to as an application gateway or forwarder) is an application that mediates traffic between a protected network and the Internet. Proxies are often used instead of router-based traffic controls, to prevent traffic from passing directly between networks. Many proxies contain extra logging or support for user authentication. Since proxies must ``understand'' the application protocol being used, they can also implement protocol specific security (e.g., an FTP proxy might be configurable to permit incoming FTP and block outgoing FTP).

Proxy servers are application specific. In order to support a new protocol via a proxy, a proxy must be developed for it. One popular set of proxy servers is the TIS Internet Firewall Toolkit (``FWTK'') which includes proxies for Telnet, rlogin, FTP, the X Window System, HTTP/Web, and NNTP/Usenet news. SOCKS is a generic proxy system that can be compiled into a client-side application to make it work through a firewall. Its advantage is that it's easy to use, but it doesn't support the addition of authentication hooks or protocol specific logging. For more information on SOCKS, see http://www.socks.nec.com/.

4- What are some cheap packet screening tools?

The Texas A&M University security tools include software for implementing screening routers. Karl Bridge is a PC-based screening router kit available from ftp://ftp.net.ohio-state.edu/pub/kbridge/.

There are numerous kernel-level packet screens, including IPf, IPfw, IPchains, pf, and IPfwadm. Typically, these are included in various free Unix implementations, such as FreeBSD, OpenBSD, NetBSD, and Linux. You might also find these tools available in your commercial UNIX implementation.

If you're willing to get your hands a little dirty, it's completely possible to build a secure and fully functional firewall for the price of hardware and some of your time.

5- What are some reasonable filtering rules for a kernel-based packet screen?

This example is written specifically for IPfwadm on Linux, but the principles (and even much of the syntax) applies for other kernel interfaces for packet screening on ``open source'' Unix systems.

There are four basic categories covered by the IPfwadm rules:

-A

Packet Accounting

-I

Input firewall

-O

Output firewall

-F
Forwarding firewall
ipfwadm also has masquerading (-M) capabilities. For more information on switches and options, see the ipfwadm man page.

5.1- Implementation

Here, our organization is using a private (RFC 1918) Class C network 192.168.1.0. Our ISP has assigned us the address 201.123.102.32 for our gateway's external interface and 201.123.102.33 for our external mail server. Organizational policy says:
Allow all outgoing TCP connections
Allow incoming SMTP and DNS to external mail server
Block all other traffic
The following block of commands can be placed in a system boot file (perhaps rc.local on UNIX systems).

```
ipfwadm -F -f
ipfwadm -F -p deny
ipfwadm -F -i m -b -P TCP -S 0.0.0.0/0 1024:65535 -D 201.123.102.33 25
ipfwadm -F -i m -b -P TCP -S 0.0.0.0/0 1024:65535 -D 201.123.102.33 53
ipfwadm -F -i m -b -P udp -S 0.0.0.0/0 1024:65535 -D 201.123.102.33 53
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0 -W eth0
/sbin/route add -host 201.123.102.33 gw 192.168.1.2
```

5.2- Explanation

Line one flushes (-f) all forwarding (-F) rules.
Line two sets the default policy (-p) to deny.
Lines three through five are input rules (-i) in the following format:
ipfwadm -F (forward) -i (input) m (masq.) -B (bi-directional) -P protocol)[protocol]-S (source)[subnet/mask] [originating ports]-D (destination)[subnet/mask][port]
Line six appends (-a) a rule that permits all internal IP addresses out to all external addresses on all protocols, all ports.
Line eight adds a route so that traffic going to 201.123.102.33 will be directed to the internal address 192.168.1.2.

6- What are some reasonable filtering rules for a Cisco?

The example in Figure 4 shows one possible configuration for using the Cisco as filtering router. It is a sample that shows the implementation of as specific policy. Your policy will undoubtedly vary.

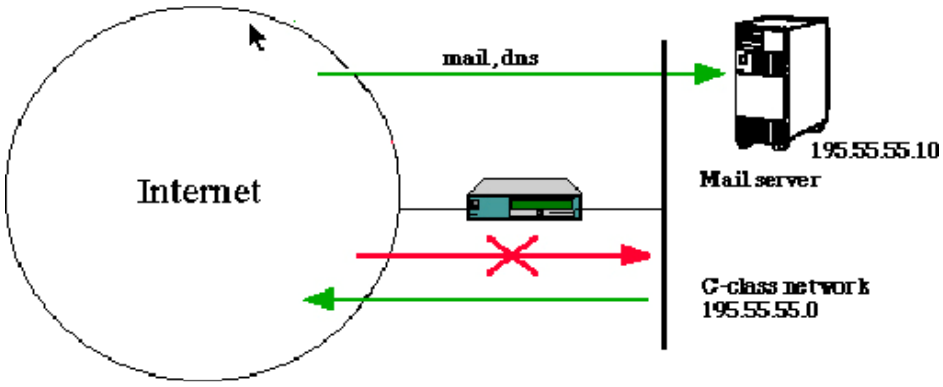

Figure 4: Packet Filtering Router

In this example, a company has Class C network address 195.55.55.0. Company network is connected to Internet via IP Service Provider. Company policy is to allow everybody access to Internet services, so all outgoing connections are accepted. All incoming connections go through ``mail host''. Mail and DNS are only incoming services.

6.1 Implementation

Allow all outgoing TCP-connections
Allow incoming SMTP and DNS to mail host

Allow incoming FTP data connections to high TCP port (> 1024)
Try to protect services that live on high port numbers
Only incoming packets from Internet are checked in this configuration. Rules are tested in order and stop when the first match is found. There is an implicit deny rule at the end of an access list that denies everything. This IP access list assumes that you are running Cisco IOS v. 10.3 or later.

```
no IP source-route
Interface Ethernet 0
IP address 195.55.55.1
No IP directed-broadcast
Interface serial 0
No IP directed-broadcast
IP access-group 101 in
Access-list 101 deny IP 127.0.0.0 0.255.255.255 any
Access-list 101 deny IP 10.0.0.0 0.255.255.255 any
Access-list 101 deny IP 172.16.0.0 0.15.255.255 any
Access-list 101 deny IP 192.168.0.0 0.0.255.255 any
Access-list 101 deny IP any 0.0.0.255 255.255.255.0
Access-list 101 deny IP any 0.0.0.0 255.255.255.0
Access-list 101 deny IP 195.55.55.0 0.0.0.255
Access-list 101 permits TCP any established
Access-list 101 permit TCP any host 195.55.55.10 eq smtp
Access-list 101 permits TCP any host 195.55.55.10 eq dns
Access-list 101 permits udp any host 192.55.55.10 eq dns
```

```
Access-list 101 deny TCP any range 6000 6003
Access-list 101 deny TCP any range 2000 2003
Access-list 101 deny TCP any eq 2049
Access-list 101 deny udp any eq 2049
Access-list 101 permit TCP any 20 any gt 1024
Access-list 101 permit icmp any
Snmp-server community FOOBAR RO 2
Line VTY 0 4
Access-class 2 in
Access-list 2 permits 195.55.55.0 0.0.0.255
```

6.2- Explanations

Drop all source-routed packets. Source routing can be used for address spoofing.
Drop directed broadcasts, which are used in smurf attacks.
If an incoming packet claims to be from a local net, loopback network, or private network, drop it.
All packets which are part of already established TCP-connections can pass through without further checking.
All connections to low port numbers are blocked except SMTP and DNS.
Block all services that listen for TCP connections on high port numbers. X11 (port 6000+), Open Windows (port 2000+) are a few candidates. NFS (port 2049) runs usually over UDP, but it can be run over TCP, so you should block it.
Incoming connections from port 20 into high port numbers are supposed to be FTP data connections.
Access-list 2 limits access to router itself (telnet & SNMP)
All UDP traffic is blocked to protect RPC services

6.3- Shortcomings

- You cannot enforce strong access policies with router access lists. Users can easily install backdoors to their systems to get over ``no incoming telnet'' or ``no X11'' rules. Also crackers install telnet backdoors on systems where they break in.
- You can never be sure what services you have listening for connections on high port numbers. (You can't be sure of what services you have listening for connections on low port numbers, either, especially in highly decentralized environments where people can put their own machines on the network or where they can get administrative access to their own machines.)
- Checking the source port on incoming FTP data connections is a weak security method. It also breaks access to some FTP sites. It

makes use of the service more difficult for users without preventing bad guys from scanning your systems.

Use at least Cisco version 9.21 so you can filter incoming packets and check for address spoofing. It's still better to use 10.3, where you get some extra features (like filtering on source port) and some improvements on filter syntax.

You have still a few ways to make your setup stronger. Block all incoming TCP-connections and tell users to use passive-FTP clients. You can also block outgoing ICMP echo-reply and destination-unreachable messages to hide your network and to prevent use of network scanners. Cisco.com use to have an archive of examples for building firewalls using Cisco routers, but it don't seem to be online anymore. There are some notes on Cisco access control lists, at least, at:

ftp://ftp.cisco.com/pub/mibs/app_notes/access-lists.

7- What are the critical resources in a firewall?

It's important to understand the critical resources of your firewall architecture, so when you do capacity planning, performance optimizations, etc., you know exactly what you need to do, and how much you need to do it in order to get the desired result.

What exactly the firewall's critical resources are tends to vary from site to site, depending on the sort of traffic that loads the system. Some people think they'll automatically be able to increase the data throughput of their firewall by putting in a box with a faster CPU, or another CPU, when this isn't necessarily the case. Potentially, this could be a large waste of money that doesn't do anything to solve the problem at hand or provide the expected scalability.

On busy systems, *memory* is extremely important. You have to have enough RAM to support every instance of every program necessary to service the load placed on that machine. Otherwise, the swapping will start and the productivity will stop. Light swapping isn't usually much of a problem, but if a system's swap space begins to get busy, then it's usually time for more RAM. A system that's heavily swapping is often relatively easy to push over the edge in a denial-of-service attack, or simply fall behind in processing the load placed on it. This is where long email delays start.

Beyond the system's requirement for memory, it's useful to understand that different services use different system resources. So the configuration

that you have for your system should be indicative of the kind of load you plan to service. A 1400 MHz processor isn't going to do you much good if all you're doing is netnews and mail, and is trying to do it on an IDE disk with an ISA controller.

Table 1: Critical Resources for Firewall Services

Service	Critical Resource
Email	Disk I/O
Netnews	Disk I/O
Web	Host OS Socket Performance
IP Routing	Host OS Socket Performance
Web Cache	Host OS Socket Performance, Disk I/O

8- What is a DMZ, and why do I want one?

``DMZ'' is an abbreviation for ``demilitarized zone''. In the context of firewalls, this refers to a neither part of the network that is neither part of the internal network nor directly part of the Internet. Typically, this is the area between your Internet access router and your bastion host, though it can be between any two policy-enforcing components of your architecture.

A DMZ can be created by putting access control lists on your access router. This minimizes the exposure of hosts on your external LAN by allowing only recognized and managed services on those hosts to be accessible by hosts on the Internet. Many commercial firewalls simply make a third interface off of the bastion host and label it the DMZ, the point is that the network is neither ``inside'' nor ``outside''.

For example, a web server running on NT might be vulnerable to a number of denial-of-service attacks against such services as RPC, NetBIOS and SMB. These services are not required for the operation of a web server, so blocking TCP connections to ports 135, 137, 138, and 139 on that host will reduce the exposure to a denial-of-service attack. In fact, if you block everything but HTTP traffic to that host, an attacker will only have one service to attack.

This illustrates an important principle: never offer attackers more to work with than is absolutely necessary to support the services you want to offer the public.

9- How might I increase the security and scalability of my DMZ?

A common approach for an attacker is to break into a host that's vulnerable to attack, and exploits trust relationships between the vulnerable host and more interesting targets.

If you are running a number of services that have different levels of security, you might want to consider breaking your DMZ into several ``security zones''. This can be done by having a number of different networks within the DMZ. For example, the access router could feed two Ethernets, both protected by ACLs, and therefore in the DMZ.

On one of the Ethernets, you might have hosts whose purpose is to service your organization's need for Internet connectivity. These will likely relay mail, news, and host DNS. On the other Ethernet could be your web server(s) and other hosts that provide services for the benefit of Internet users.

In many organizations, services for Internet users tend to be less carefully guarded and are more likely to be doing insecure things. (For example, in the case of a web server, unauthenticated and un trusted users might be running CGI, PHP, or other executable programs. This might be reasonable for your web server, but brings with it a certain set of risks that need to be managed. It is likely these services are too risky for an organization to run them on a bastion host, where a slip-up can result in the complete failure of the security mechanisms.)

By putting hosts with similar levels of risk on networks together in the DMZ, you can help minimize the effect of a break-in at your site. If someone breaks into your web server by exploiting some bug in your web server, they'll not be able to use it as a launching point to break into your private network if the web servers are on a separate LAN from the bastion hosts, and you don't have any trust relationships between the web server and bastion host.

Now, keep in mind that this is Ethernet. If someone breaks into your web server, and your bastion host is on the same Ethernet, an attacker can install a sniffer on your web server, and watch the traffic to and from your bastion host. This might reveal things that can be used to break into the bastion host and gain access to the internal network. (Switched Ethernet can reduce your exposure to this kind of problem, but will not eliminate it.)

Splitting services up not only by host, but by network, and limiting the level of trust between hosts on those networks, you can greatly reduce the

likelihood of a break-in on one host being used to break into the other. Succinctly stated: breaking into the web server in this case won't make it any easier to break into the bastion host.

You can also increase the scalability of your architecture by placing hosts on different networks. The fewer machines that there are to share the available bandwidth, the more bandwidth that each will get.

10- What is a `single point of failure', and how do I avoid having one?

An architecture whose security hinges upon one mechanism has a single point of failure. Software that runs bastion hosts has bugs. Applications have bugs. Software that controls routers has bugs. It makes sense to use all of these components to build a securely designed network, and to use them in redundant ways.

If your firewall architecture is a screened subnet, you have two packet filtering routers and a bastion host. (See question 3.2 from this section.) Your Internet access router will not permit traffic from the Internet to get all the way into your private network. However, if you don't enforce that rule with any other mechanisms on the bastion host and/or choke router, only one component of your architecture needs to fail or be compromised in order to get inside. On the other hand, if you have a redundant rule on the bastion host, and again on the choke router, an attacker will need to defeat *three* mechanisms.

Further, if the bastion host or the choke router needs to invoke its rule to block outside access to the internal network, you might want to have it trigger an alarm of some sort, since you know that someone has gotten through your access router.

11- How can I block all of the bad stuff?

For firewalls where the emphasis is on security instead of connectivity, you should consider blocking *everything* by default, and only specifically allowing what services you need on a case-by-case basis.

If you block everything, except a specific set of services, then you've already made your job much easier. Instead of having to worry about every security problem with everything product and service around, you only need to worry about every security problem with a specific set of services and products.

Before turning on a service, you should consider a couple of questions:

- Is the protocol for this product a well-known, published protocol?
- Is the application to service this protocol available for public inspection of its implementation?
- How well known is the service and product?
- How does allowing this service change the firewall architecture? Will an attacker see things differently? Could it be exploited to get at my internal network, or to change things on hosts in my DMZ?

When considering the above questions, keep the following in mind:

- ``Security through obscurity'' is no security at all. Unpublished protocols have been examined by bad guys and defeated.
- Despite what the marketing representatives say, not every protocol or service is designed with security in mind. In fact, the number that is is very few.
- Even in cases where security is a consideration, not all organizations have competent security staff. Among those who don't, not all are willing to bring a competent consultant into the project. The end result is that otherwise-competent, well-intended developers can design insecure systems.
- The less that a vendor is willing to tell you about how their system *really* works, the more likely it is that security (or other) problems exist. Only vendors with something to hide have a reason to hide their designs and implementations [2].

12- How can I restrict web access so users can't view sites unrelated to work?

A few years ago, someone got the idea that it's a good idea to block ``bad'' web sites, i.e., those that contain material that The Company views ``inappropriate''. The idea has been increasing in popularity, but there are several things to consider when thinking about implementing such controls in your firewall.

- It is not possible to practically block everything that an employer deems ``inappropriate''. The Internet is full of every sort of material. Blocking one source will only redirect traffic to another source of such material, or cause someone to figure a way around the block.
- Most organizations do not have a standard for judging the appropriateness of material that their employees bring to work, e.g., books and magazines. Do you inspect everyone's briefcase for ``inappropriate material'' every day? If you do not, then why would you inspect every packet for ``inappropriate material''? Any

decisions along those lines in such an organization will be arbitrary. Attempting to take disciplinary action against an employee where the only standard is arbitrary typically isn't wise, for reasons well beyond the scope of this document.

- Products that perform site-blocking, commercial and otherwise, are typically easy to circumvent. Hostnames can be rewritten as IP addresses. IP addresses can be written as a 32-bit integer value, or as four 8-bit integers (the most common form). Other possibilities exist, as well. Connections can be proxies. Web pages can be fetched via email. You can't block them all. The effort that you'll spend trying to implement and manage such controls will almost certainly far exceed any level of damage control that you're hoping to have.

The rule-of-thumb to remember here is that you cannot solve social problems with technology. If there is a problem with someone going to an ``inappropriate'' web site, that is because someone else saw it and was offended by what he saw, or because that person's productivity is below expectations. In either case, those are matters for the personnel department, not the firewall administrator.