

Running Head: Considerations in Choosing a Firewall

Considerations in Choosing a Firewall

Daniel Owen

spam@danielowen.com

<http://www.danielowen.com>

Abstract

This paper looks at the available firewall technologies in current use. Both advantages and disadvantages for each technique are discussed. Techniques reviewed include packet filtering, proxies, stateful inspection and deep packet inspection. Also discussed are combinations of techniques and defense in depth. Along with these factors other important factors such as management interfaces, hardware choices and build verses buy are discussed.

In the early days of networked computing most systems were open and accessible. It could even be argued that the engineers went out of their way to see to it that systems were freely accessible. During these early days security was not a major concern for most computer scientists or network designers. As local networks were connected to other networks this initial open architecture allowed for many people to quickly and easily communicate. Unfortunately it also allowed people with ill intent to attack these systems. Since early networked systems had been built without security in mind many early systems had numerous security vulnerabilities inherent in their design.

As more networks were connected it became obvious to network and system administrators that it was necessary to find a way to protect their systems from rogue operators with ill intent. As a first solution to fill this gap, engineers started to filter network traffic using edge routers that separated protected networks from public or hostile networks. These early systems were simple static packet filters that allowed or denied specific types of traffic based on information found in packet headers. This simple approach was a huge step in the direction of more secure networks. Applications and operating systems with inherent security flaws could be separated from the outside world to provide them with a previously unavailable level of security. Firewalls have evolved from these simplistic packet filtering routers into the complex systems that they are today.

Security problems similar to those that lead engineers to implement static filtering on routers still exist today. The exploits that attackers use are more complex as are the security vulnerabilities attackers are exploiting but the fact remains that we are no closer to systems that are secure out of the box today than we were when the first network

administrator started filtering border network traffic to block attacks coming from connected networks. As long as this fact remains there will be a need to firewalls as part of an overall security architecture.

Modern firewalls are similar in purpose to their early counterparts. In the simplest terms all network firewalls serve one purpose. They separate one network from another much like a firewall in a building separates the building into contained building sections in the event of a fire. By separating networks the risk of an attacker from one network successfully infiltrating the protected network is reduced. The most common location for a firewall is separating a LAN from the Internet. Since research shows that a large portion of network intrusions are carried out by insiders, companies are beginning to separate their LAN segments using internal firewalls in much the same way that they protect their LAN from the hostile Internet.

Even though firewalls do provide a great deal of protection they are not a panacea. By definition networks are connected to allow the flow of data between networks. These legitimate channels of communication can provide an avenue of attack. For example, many companies will have a firewall protecting their web server that only allows traffic on port 80 to pass. This is a fairly strong position since it means that any attack that relies on attacking services not listening on port 80 will not be able to make a connection to even attempt a compromise. On the other hand if the web server software has a vulnerability most firewalls will allow the attack traffic through and the server may be compromised. This is beginning to change with some newer firewall technologies that will be discussed later, but many firewall implementations are vulnerable to this type of attack. Firewalls are also not able to protect against internal users who may attack

systems. Further the most difficult attack to protect against is legitimate users using their access for illegitimate purposes. Firewalls are helpless against these types of attacks. An attack that can disguise itself to look like legitimate traffic will be able to bypass many firewalls.

While these shortcomings may sound like firewalls are no longer a useful tool the opposite is true. Firewalls will protect against numerous attacks and make anyone trying to launch an attack work harder. Since a large proportion of attackers are looking for easy targets firewalling is very effective against this type of casual attack. At the same time it is important to understand what can and can not be expected from a firewall. This will allow a security decision maker to evaluate additional precautions that may need to be taken in addition to a firewall.

As a general rule a firewall is still the best first line of defense for most companies. The level of required security will determine the necessary additional security layers added in addition to the firewall. Until we have ubiquitous secure operating systems firewall solutions will continue to be the first line of defense for most companies' security architectures. As the technology improves some of the problems that current firewalls can not protect against will be defendable with firewalls while others will always require other approaches to security.

Basic Types of Firewalls

There are a number of overlapping types of firewalls. These start with the oldest packet filtering routers and make a progression to our current state of the art firewall technology. Each generation of firewall developer learns from and improves on the generation that came before them. By this continual evolution each generation of firewall

is able to protect against more threats than the generation the preceded it. This does not mean that every business needs the newest generation of firewall technology or that the older technologies do not have a place in the modern security architecture. Each type of firewall technology has inherent advantages and disadvantages that will be reviewed.

All firewall technologies can essentially be broken into two categories. Packet filtering firewalls are relatively simplistic and usually look at traffic at the network level and below. On the other hand application proxies look at traffic throughout all seven layers of the OSI model. Many of the technologies that will be discussed blur this distinction.

Static Packet Filters

The simplest form of packet filter is the static packet filtering router. A static packet filter looks at the packets coming in or leaving the network and based on a set of rules decides whether to allow that transaction to take place. Static packet filters are without question the simplest filters. Static packet filters have a simple yes or no response based on information in the headers such as origination or destination address, ports and connection flags for every packet that travels through the router. (Desal, 2002)

Static packet filters were a great improvement over leaving servers exposed to connected networks, but they do have shortcoming that have become more problematic as attackers become more sophisticated. Static packet filters are rather simplistic in their view of traffic since they only look at information in the headers for packets but not the data. Due to this simplicity a number of trivial attacks can be initiated to spoof traffic thus allowing unwanted traffic through the firewall.

FTP is a major problem for static packet filters. FTP operates differently from most protocols. FTP listens on port 21 for new connections. Once it receives a connection it can have two different procedures for responding. These are referred to as active and passive mode. In active mode the client connects to port 21 and when data is to be transferred the client allocates a random port above 1024 that the server connects to from port 20. In passive mode the connection to port 21 by the client is the same but the server allocates a port above 1024 for the data connection. All ports above 1024 must be opened on either the client or server side respectively to allow this second connection since a port is chosen at random (Robertson, 2004). This is not acceptable for most organizations since it leaves thousands of high number ports open for exploit.

A similar problem is that connections must be allowed in both directions. So if we want to allow people to connect to web sites on the internet two rules must be created. One rule is needed to allow traffic out with another rule allowing response traffic back into the network. Since these are static connections an attacker can easily connect to internal clients and services by spoofing traffic reporting to be a response to a request from an internal connection.

Static packet filters do have some inherent advantages due to their simplicity. Due to the simplicity of static packet filters they can be implemented using very minimal additional resources on the equipment they are added to with little if any measurable increase in latency. This simplicity makes packet filtering the least expensive form of firewall technology. Most modern routers will at a minimum have packet filtering software bundled with them.

The shortcomings of packet filtering along with price drops in other technologies have essentially ended the packet filter as a single line of defense for most companies. This does not mean that static packet filters are not used in modern organizations. They are still useful in a few circumstances. Static packet filtering can be used to protect network equipment. When security vulnerabilities are found in networking equipment a common temporary solution until patches become available is blocking the ports that the attack uses. Due to the speed of static packet filtering many businesses will block specific unwanted traffic at the router in an effort to help eliminate overhead for the firewall. Even though static packet filtering is not typically an end in itself any longer it still can be useful as an additional layer of security.

Static packet filters are not sold as stand alone firewalls today but understanding static filtering will make other filtering technology easier to understand and static filters are still often included as an option in router configurations. Packet filtering is essentially a free value added feature in today's market.

Application Proxy

Application proxies were developed to help eliminate many of the shortcomings inherent in early packet filters and they were successful on many levels.

Application proxies sit between the protected network and an untrusted network like any other border firewall. They work in much the same way that a trusted proxy would work in the physical world. That is to say, the proxy accepts connections on behalf on the client that it is protecting and then the proxy creates a new connection to the client at which time it delivers the original message. This eliminates the need to allow direct connections from untrusted clients to trusted servers.

This approach can potentially have a much higher level of security than packet filtering. Since the entire packet is being delivered to the proxy server the entire packet through layer seven of the OSI model can be inspected. This high level of inspection allows for the ability to detect malicious or malformed elements above the network layer. This being said, how thoroughly packets are actually inspected depends on the proxy server software being used. Because of this it is important to review what is being inspected when evaluating proxy software.

To go along with this higher level of security proxy servers bring their own new challenges to the table. First of all due to the much higher level of processing being done proxy servers are much more resource intensive in both processing power and memory requirement. This can be a problem for larger networks even using today's fast hardware. Further, due to the higher level of inspection some level of additional latency can be caused by application proxies. Due to these resource considerations proxies may not be practical for some applications, such as voice over IP, that require very low latency.

Possibly the biggest downside to using a proxy is that each application needs to have a proxy written for it. In other words a proxy server that will support SMTP, HTTP and FTP will need to load three separate proxies. For common protocols such as the three mentioned this is not normally a problem, but in the case of newer applications, unusual applications or custom applications proxies may not be available. This can be a rather large problem for companies that use a great deal of custom applications or adopt new technology quickly.

Proxies fall into two categories. There are manual setup proxies and transparent proxies. A manual setup proxy requires either special configuration of proxied

applications or special software to be loaded on the client machine. From a network configuration standpoint this is the easiest configuration but also the most intrusive since users or support staff must configure each client to use the proxy. Since a manual step is required this is only practical for protecting internal clients connecting out to a hostile network, such as the Internet, or known clients connecting to a server in the protected network. Transparent proxies essentially are invisible to the end user. This allows for a much simpler end user configuration at the expense of a more complex initial proxy server configuration. A type of transparent proxy called a reverse proxy can be set up to protect internal servers such as web servers from traffic originating from an untrusted network.

Application proxies range in price depending on capacity requirements. Much of the cost involved in application proxies is hardware dependent. Unlike packet filtering that can run on relatively inexpensive hardware application proxies typically must have fast processors and a large amount of ram.

There are two final features that proxy servers almost have a monopoly on that are not specifically security measures but are worth mentioning. Most proxy server implementations can be configured to cache content that it has previously retrieved. This can be an advantage for companies that have limited Internet bandwidth since popular sites will only need to be retrieved once in a given time range rather than for each person who wants to use the site.

Proxies can be set up to stop access to undesirable content. This application of a proxy can be used as a security measure but more often than not is used for other reasons. This can include blocking unproductive sites such as sites containing jokes or sports

scores, sites that serve advertising or it can be used to block content such as pornography that may create legal difficulties for the company. It is also possible to block known sites that distribute malicious software or spyware.

Stateful Inspection

Stateful inspection, also referred to as dynamic packet filtering, was introduced in 1993 by Check Point Software Technologies, Inc. in their Firewall-1 product (Webopedia n.d.). Within a short period of time stateful inspection became the predominant type of packet filtering firewall. The reason for this is quite simple. Stateful inspection alleviated many of the shortcomings of basic static packet filtering without introducing the overhead of an application proxy.

Stateful inspection is the next logical step in packet filtering. It builds on the low overhead of static packet filtering while allowing for more granular control and a much higher level of security. Stateful inspection adds the concept of watching connection state to static packet filtering. The firewall watches each TCP connection as it is being set up and keeps up with its status until the session either times out or is torn down by the application using the connection. Stateful Inspection firewalls use virtual connections for connectionless protocols such as UDP and RDP. Tracking session state eliminates the spoofing problems associated with static filters. Since connections are watched as they are set up and torn down the need for opening large numbers of ports to support FTP is also eliminated. (Check Point Software Technologies Ltd., 2004, Stateful inspection.)

Due to these improvements over static filtering most port filtering has moved to stateful inspection. Many newer routers have implemented stateful inspection as a

replacement for static filters due to the limited additional overhead and much improved security.

Stateful inspection firewall packages are at all ends of the spectrum in ease of use and price. Some stateful inspection firewalls use a pure command line interface similar to that of a stateless packet filter while others use very advanced GUI interfaces that allow even a novice administrator to configure the firewall as long as they have an understanding of the protocols that need to be allowed to travel through the firewall.

Stateful inspection firewalls vary in price depending on desired functionality and the size of the network to be protected. There are a number of sub \$100 stand alone firewall on the market that are geared toward the SOHO environments. At the other extreme firewalls designed to protect large networks can easily go into the tens of thousands of dollars. Stateful inspection firewall software is also build into OpenBSD, FreeBSD, NetBSD and Linux which are all free after the cost of hardware and implementation.

Deep Packet Inspection

Deep packet inspection, also referred to as application intelligence, is a compromise between the best points of stateful inspection and a proxy. Deep packet inspection starts with simple stateful inspection but it has the ability to inspect the entire network stack to look for attacks in supported protocols. Deep packet inspection is a natural outgrowth of the weaknesses of stateful inspection. Since stateful inspection is unaware of what is going on inside the data of packets many application layer attacks that can be stopped by a proxy are allowed through by stateful inspection. The most common examples of this are malicious web traffic being used to attack a web server and web

servers hosting malicious content that will be downloaded by regular web users. Since these are both examples of traffic going over open ports the traffic will be allowed through by a stateful inspection firewall. In the case of a malicious web site this is traffic being allowed through because it is a response to a legitimate request from inside the protected network. With a stateful inspection approach this type of malicious traffic will be allowed by design. (F5, October 2004)

Some stateful inspection implementations, such as Check Point's Firewall-1, have had simple application proxies built into their products for years. These have helped to protect the most common attack vectors but were by design limited to only a select number of high profile targets. These proxies were also not as full featured as deep packet inspection. They did not look for specific attack signatures. These early proxies built into stateful inspection firewalls looked at simple RFC violations or common generic buffer overflow issues instead of specific attack signatures. (Check Point Software Technologies Ltd. 2004, Check Point application intelligence)

Deep packet inspection takes this rudimentary inspection and goes a step further. Not only are simple overflows and RFC violations detected but known attack vectors are also defended. This allows administrators to protect known and unknown vectors of attack while not taking on all of the higher processing requirements of traditional proxies. (F5, October 2004)

Deep packet inspection does have some issues that must be considered when evaluating technologies. Some of these issues carry over from stateful inspection and proxies while others are unique to deep packet inspection.

First of all there is the issue of processing power. Deep inspection does not typically require the level of procession power required by a proxy due to its more limited scope, but an added level of processing load is imposed. This may not be a problem for most firewalls built on PC server architecture, but it may be a problem for solutions that are built on top of specialized hardware or a device with limited processing power such as a router. The processing requirements and continual evolution of the attacks looked for will also make it more difficult to gain speed by using ASICs as has been the case in early deep packet inspection solutions form Netscreen (Roberts, 2003).

Since some of the added security is signature based, there is also an issue of delay in protection from new attacks. Since much of the improvement offered by deep packet inspection relies on looking for specific attack signatures there is an inherent delay between a new attack being identified and a vendor providing an inspection signature. This time will vary from vendor to vendor and should be a consideration when looking at deep packet inspection firewalls. These updated signatures must also be downloaded and installed so ease of updates must also be considered.

Overall deep packet inspection is a security improvement over stateful inspection while still allowing faster processing than a traditional application proxy. For many companies this compromise will be an ideal middle ground between stateful inspection and a proxy.

Combinations

Many products use multiple firewall techniques. This has the advantage of allowing the product to take the best features of multiple techniques while trying to downplay the weaknesses inherent in a given design. A perfect example of this is deep

packet inspection which to a great extent is the marriage of stateful inspection and proxy technology. Deep packet inspection may not have as full a feature set as some stand alone proxy implementations, but it also does not have all of the disadvantages of a traditional proxy solution.

One thing that must be watched for in promotional literature is companies mixing technologies without making it clear that they are doing so. Stateful inspection is one place that this is especially true. Some companies have added limited applications proxies into products that they simply refer to as stateful inspection. Just because some companies include specific features a customer should not assume that a product claiming to use a specific inspection technique does any more than the minimum functionality inherent in that technology. Comparing products that include additional functionality but use the same naming can be difficult without doing a thorough investigation of all product capabilities.

Defense in Depth

Another method that some companies use to improve their perimeter security is defense in depth. Defense in depth is not exclusively a perimeter security concept but for the purposes of this paper I will only look at perimeter security implementations. A typical example of perimeter defense in depth follows. First the company uses a border router to stop very simple network attacks. Then traffic is passed to a stateful inspection firewall that determines whether to allow traffic based on a more complex rule set. Finally the firewall passes specific traffic to a proxy server located in the DMZ that proxies for commonly attacked applications such as web and mail servers.

This approach allows for the simplest attacks to be stopped by simple, relatively fast machines and thus reducing latency and cost while still providing the highest possible level of perimeter security. The obvious disadvantage of this approach is that it requires that multiple devices and rule sets be maintained. This can add to both complexity and cost. For the most part the level of required security will determine the level of defense in depth.

Intrusion Prevention Systems

Intrusion Prevention Systems (IPS) are not firewalls but they do have some overlap and as such I believe they must be mentioned briefly. IPS is an outgrowth of Intrusion Detection System (IDS) research. IDS and IPS both try to look for attack signatures and either log or block the attack respectively. There is some overlap in the goal of deep packet inspection and IPS but the methodology and size of attack databases are generally different.

IPS systems typically have two methods of learning about attack signatures. The simplest method is periodic updates of known attack vectors. This closely mirrors the method used for updating deep packet inspection systems. The second method of learning is to watch a network and learn what is normal and then begin to block traffic that does not statistically fit the learned patterns. (Franklin, 2005)

IPS systems as a general rule will have a larger set of known attack vectors. IPS systems will also have a higher likelihood of having false positives and blocking legitimate traffic due to their more aggressive stance toward blocking malicious traffic. For some companies the added risk of blocking legitimate traffic is justified by the

additional security offered by an IPS system. (Franklin, 2005) IPS is another level of defense in depth, but it is not a replacement for the traditional firewall.

Applications Firewalls

Application firewalls have some similarities to IPS systems in that they try to look at normal application behavior and then allow or deny traffic based on whether a given request meets the expected normal behavior. Application firewalls have for the most part been implemented for web applications. Much like IPS applications firewalls can rely on either signatures of known attacks or learn normal traffic behavior and block unusual traffic (Bar-Gad, 2002). Applications firewalls are not a replacement for the border firewall but an additional level of defense in depth for specific high risk applications.

Client Firewalls

So far I have concentrated on network level firewalls that sit at the border between a trusted and an untrusted network. This is the easiest place to implement a firewall and arguably provides the highest level of security for the expenditures required. The down side to this type of implementation is it creates a crunchy exterior to protect the network from the outside world but if an attacker penetrates the outer fortification the internal resources are open for exploit. This is especially problematic in situations where many companies share internal networks such as collocation facilities. The crunch exterior model can also open a company up to insider abuse.

The answer to this problem has been implementation of client level firewalls. Client firewalls can range from very simple port filters like the ones available in Windows NT and Windows 2000 all the way up to fully functional stateful inspection as

is included in some UNIX like operating systems. There are also numerous add on client level stateful inspection firewalls available for a large number of operating systems as well as hardware firewalls built into network interface cards. This can be a useful addition to defense in depth or a replacement for a border firewall if only one or two machines need to be protected.

Management Considerations

After deciding on desired firewall architecture there are a number of considerations that must be undertaken before a final product decision is made. These considerations include ease of use, build verses buy and hardware architecture.

Ease of Use

The biggest issue that will affect all firewall implementations regardless of size is the firewall management interface. In the broadest terms there are two choices both with their advantages and disadvantages. There are GUI based firewall systems and there are command line driven systems.

Command line driven firewalls typically approach rule base editing assuming that the person making changes to the rule base knows what they are doing and will make sure they do not make any changes that might lead to unexpected outcomes. With this assumption comes great flexibility at the expense of a safety net to keep an administrator from making changes that might be detrimental to the overall security provided by the firewall. For many administrators this level of control is desired. For less experienced or careless administrators this is simply asking for trouble. Many products, such as Cisco

PIX, iptables, ipfilters and others that are command line driven have add on GUI clients available (Kurland, 2005) for those who prefer a GUI client for day to day administration.

A GUI based system will be easier for most administrators to manage because it will allow a graphical representation of the rule set that can be easily changed with a point and click or web based interface. GUI interfaces vary widely in their implementation and their abilities, so it is advisable for anyone looking at a GUI driven firewall to try the GUI management client before purchasing. A GUI can limit the options available to the user. While this does have the advantage of keeping an inexperienced or careless administrator from breaking their firewall in unexpected ways it may also keep an experienced administrator from being able to make certain changes that are required for their unique network.

To overcome the shortcomings of the GUI some products may allow limited access to the firewall rules through a command line interface. As an example Check Point's Firewall-1 product has an award winning GIU but some specialized changes to the firewall require editing that can not be done through the GUI. For these changes Check Point provides a tool that allows editing of the rule base and configuration files from the command line. Other systems with a GUI interface may allow editing of configuration files in a text editor for unusual changes that are difficult or undesirable to implement in a GUI. This is an ideal solution that takes advantage of the ease of a GUI for that majority of changes but allows the flexibility of a command line editor when it is needed. When choosing a solution that uses a GUI for administration, it is advisable to investigate the ability to make changes outside of the GUI. A GUI can become very restrictive for a system that only allows changes to be made using the GUI.

Other Management Features

For large installations that require multiple firewalls for scalability or failover, keeping all firewalls in synchronization can become an issue. For installations that will require multiple firewalls it is advisable to look for a firewall solution that has a management interface that allows for editing of all firewalls from a single centralized administrative console. This will make management easier and help to reduce the risk of multiple firewalls protecting the same assets being configured differently.

In a single firewall implementation the firewall becomes a single point of failure. For many companies this is an acceptable risk. This is especially true if there are other single points of failure, but if a single point of failure is considered an unacceptable risk the firewall solution chosen should offer instant fail over across multiple devices. This functionality is available for many enterprise level firewalls. When comparing firewalls it is worth looking at what will happen to sessions that are already initiated when a firewall fails. Different vendors will have different solutions for this problem.

Finally, support options must be considered. Most commercial firewall solutions will offer some type of support but this support can be costly. As an example, Check Point software charges twenty to thirty percent of the current product cost per year for ongoing regular business hours support and upgrades (Check Point Software Technologies Ltd., June 29, 2004, Enterprise Software Subscription & Standard Support Program). If a customer needs twenty-four hour a day support the price increases to as much as forty percent of the current purchase price for the supported software (Check Point Software Technologies Ltd., June 29, 2004, Enterprise Software Subscription & Premium Support Program). A customer always has the option of not purchasing

upgrades and support but that essentially means that if they need any type of support they can not turn to the vendor for support. This includes issues caused by software defects. This is not a situation that most companies want to be in with a product as complex and crucial as a firewall.

Support contracts for firewall maintenance can be expensive but for many companies the cost of the contract is money well spent to insure that there is someone responsible for supporting the product. At the other extreme, the open source movement offers software with no upfront or support contract costs but all support functions must be handled by internal staff without the assistance of vendor engineers that can be called on for support. For some open source products third party companies can be contracted for support. As a general rule the open source community does try to help with problems, but for problems that are unusual finding someone who can help often involves finding someone who has already had the same problem you are experiencing and has found a solution.

While support contracts can be costly the final decision on vendor support requirements comes down to looking at the organization's requirements and deciding what level of vendor support is necessary. For some companies e-mail with a turnaround time of several hours or even days may be acceptable. For other companies telephone support with a response in a matter of minutes may be the only acceptable level of support. The decision comes down to choosing the solution that is best for a given company and entering into that solution knowing up front what the long term supports cost and expectations will be.

Build Verses Buy

When deciding on a firewall solution deciding whether to build a firewall server or buy a complete system is a very important decision. The decision will come down to evaluating several factors that will determine what is appropriate each company's unique circumstances.

Purpose Built Solutions

The simplest choice is to purchase a complete solution that includes both hardware and software. With a black box solution all that is typically involved in setup is plugging the device in between the trusted and untrusted networks and configuring a rule base.

With this ease of use comes a certain level of lost upgrade flexibility. With a closed solution the only way to upgrade may be replacement of the entire device. For this reason when selecting a complete hardware based solution it is important to consider future growth and the ability to expand and upgrade both the hardware and software that comprises the firewall.

From a cost standpoint complete solutions are generally mixed depending on the requirements of the company purchasing the firewall. In the home and SOHO market there are numerous products available in the sub \$100 range and many of the larger established firewall companies have enter the market with products designed for telecommuters and branch offices that can be managed remotely. These products typically retail in the sub \$500 range. This is less expensive than the cost of purchasing hardware to build a firewall. In larger implementations the price difference is less of an issue and the decision usually comes down to choosing a feature set that best meets the

needs for a given company. Many people appreciate the ability to have one person to call for all support issues as opposed to having separate firewall software, operating system and hardware vendors that may blame each other for defects.

One final advantage that some purpose built solutions have is that the software can be burned into the hardware allowing for better performance. Not all manufacturers of purpose built systems use ASICs but in situations that require very high throughput with minimal delay solutions that take advantage of ASICs can have a substantial performance advantage. On the other hand these systems can be impossible to upgrade due to their specialized nature.

Commercial Software

Another popular option is purchasing a commercial application that can be loaded onto preexisting hardware architecture. These solutions can fall into two camps. The first and easiest group involves the firewall vendor shipping a preconfigured hardened operating system along with the firewall software. This option has an ease of implementation advantage similar to buying an integrated hardware solution at the cost of limitations in changes that can be made to the underlying operating system.

The second and by far more common approach is selling a firewall as an application that is loaded on top of a general purpose operating system such as Linux, Solaris or Windows. This has the advantage of flexibility. As long as the licensing allows it, the firewall software can be loaded on any type of general purpose hardware that is currently available in the market. If a company feels that the firewall is running too slowly the underlying operating system or hardware can be tuned to improve performance without having to replace the entire firewall. If a company needs to have

numerous network segments separated by the firewall the only limiting factor is the number of ports supported by the hardware and underlying operating system.

Two additional challenges come with this flexibility. The first challenge is complexity. The firewall administrator is no longer only managing a firewall but now is also administering the underlying general purpose operating system and hardware. Secondly, the underlying operating system must be hardened so that it does not provide security vulnerabilities. The second issue is mitigated to some extent by the firewall software since most firewall packages are designed to hook at a very low level in the network stack, but vulnerabilities in the host operating system must be taken into consideration as a possible attack vector when loading firewall software on top of a general purpose operating system.

The more secure the underlying operating system the better security that the firewall will be able to provide. It is equally important when choosing an operating system to consider the expertise available within the company. An administrator who is not familiar with an operating system will have difficulty in managing the system when inevitable problems occur. Using an operating system that in house administrators are familiar with will also allow administrators to do a better job of hardening and tuning the underlying operating system. As a general rule the advice most often given by experts is to chose what you are most familiar with and allow the firewall to do its job in handling the operating system vulnerabilities after you have hardened the operating system as much as is possible.

The cost of commercial software varies widely depending on the vendor chosen, the number of protected clients and the required feature set. As a general rule for a small

number of clients using a fairly simple firewall the price can run in the low hundreds of dollars range. For a large company protecting thousands of clients and using advanced features the price can go into the hundreds of thousands of dollars. Even with similar functionality the prices can vary a great deal between vendors.

Open Source Solutions

For the most part, open source solutions have many of the same advantages and disadvantages as loading a commercial firewall application onto a general purpose operating system. The major differences are cost, flexibility and support options.

Open source solutions are generally speaking free of up front software purchase costs. That does not necessarily mean that they are free. To a great extent the overall cost depends on a company's staff expertise. For a company that has staff that have experience with one or more open source operating systems the learning curve for the built in or add on firewalls available should be fairly shallow. On the other hand, for a company that does not have in house expertise consulting or training costs can quickly reach a point that makes open source software prohibitively expensive to implement.

Open source proponents pride themselves on the flexibility of their software. This flexibility follows through into the area of firewalls. In the open source area there are numerous firewalls to choose from all with different feature sets. Unlike most commercial firewalls many open source products are designed to allow for multiple products to interoperate in a way that allows a site to customize a group of applications to their own unique needs, and if a feature is not available a company is open to customize the existing software to make it work for their needs.

Open source support is another area that is rather mixed. There is not a single entity that can be forced to fix problems, but the maintainers of most active open source projects want to see their project grow so it is in their interest to see that known bugs are fixed as quickly as possible. For general management support there are numerous small and large companies that offer maintenance contracts for open source software.

Conclusion

The specific vulnerabilities that must be protected continue to evolve with time, but the need to protect internal networks from more hostile networks does not change. To a great extent, the history of firewall technology has been an arms race. As attackers become more advanced the tools used to protect networks must advance at the same rate. As old attack vectors are protected attackers find new methods of attacking protected networks.

Advances have lead to a number of changes in the area of firewall technology that make them more capable of repelling attackers. Companies must look at their unique security stance and decide how much perimeter security is enough. For some companies this may be a basic stateful inspection firewall that has only two or three rules. For other companies the only appropriate solution may be to use several advanced firewalls with complex rule sets configured to provide defense in depth.

In the modern world there are a number of choices in firewall technology but at the most basic level they all fall into two basic types packet filters and proxy servers. These techniques have evolved and expanded into numerous overlapping technologies but understanding the two original firewall techniques is the key to understanding all future techniques that build on them.

A firewall will not fix all networks security problems, but it will go a long way toward helping to protect a network. Firewalls can not protect against abuse of legitimate services. For this type of protection systems must still be protected in other ways. What a firewall will do is protect from certain attacks originating from hostile networks and in some cases they can be used to buy time in protecting other services.

A properly configured perimeter firewall is the basic first line of defense for most companies. If this line of defense is carefully chosen and implemented it will, arguably, go farther than any other single defensive measure that can be implemented. For this reason firewalls are the most commonly implemented single line of defense and as such choosing the firewall that is right for a specific business is of utmost importance.

References

Bar-Gad, I. (Jun 3, 2002). Web application firewalls protect data. *Network World*, 19(22), 47.

Check Point Software Technologies Ltd. (2004). Check Point application intelligence. Retrieved February 9, 2005, from http://www.checkpoint.com/products/downloads/applicationintelligence_whitepaper.pdf.

Check Point Software Technologies Ltd. (2004). Stateful inspection. Retrieved February 14, 2005, from www.checkpoint.com/products/downloads/Stateful_Inspection.pdf

Check Point Software Technologies Ltd. (June 29, 2004). Enterprise Software Subscription & Standard Support Program: Terms and Conditions. Retrieved April 10, 2005, from http://www.checkpoint.com/techsupport/programs/docs/standard-terms_conditions.pdf

Check Point Software Technologies Ltd. (June 29, 2004). Enterprise Software Subscription & Premium Support Program: Terms and Conditions. Retrieved April 10, 2005, from http://www.checkpoint.com/techsupport/programs/docs/premium-terms_conditions.pdf

Desal, M. S., Richards, T. C. & Von der Embse T. (2002). System insecurity – firewall. *Information Management & Information Security*, 10(2/3), 135-139.

F5. (July 2004). Web application vulnerabilities and avoiding application exposure. Retrieved February 2, 2005, from http://www.f5.com/f5products/products/TrafficShield/Vulnerabilities_White_Paper.pdf.

F5. (October 2004). TrafficShield application firewalls. Retrieved February 2, 2005, from <http://www.f5.com/solutions/tech/TrafficShield.pdf>.

Franklin, C. Jr. & Weins, J. (January 20, 2005). Intrusion-protection systems the great IPS test. *Network Computing* (pp 55-68). Retrieved February 6, 2005 from

http://i.cmpnet.com/nc/1601/graphics/1601f3_file.pdf.

Kurland, V. (April 3, 2005) Frequently Asked Questions for Firewall Builder 2.0.

Retrieved February 10, 2005 from

http://www.fwbuilder.org/archives/cat_faq.html.

Roberts, P. (October 20, 2003). NetScreen announces deep inspection firewall.

NetworkWorld. Retrieved February 9, 2005 from

<http://www.nwfusion.com/news/2003/1020netscannou.html>.

Robertson, P., Matt Curtin, M. & Ranum, M. J. (July 26, 2004). Internet firewalls:

frequently asked questions (Version 10.4). Retrieved February 14, 2005, from

<http://www.interhack.net/pubs/fwfaq/>.

Webopedia. (n.d.). What is stateful inspection?. Retrieved February 14, 2005, from

http://www.webopedia.com/TERM/S/stateful_inspection.html.