Global Information Assurance Certification Paper

Case Study: Internet Firewall Architecture

Author: Preston Wade
Date: Jan 13 2002
Assignment: V 1.4b Option # 2

Table of Contents

Abstract

We had built our Internet firewall architecture around application level gateway-type firewalls. At the time we were only using roughly a T1's worth of bandwidth so we felt like performance of the firewall wouldn't be an issue. Over time this and other issues would prove that the application level gateways would not provide the best solutions for the company's business needs.

After a few years of mergers and acquisitions it was clear that something needed to change with our existing Internet firewall architecture. The Internet firewalls were quickly becoming the bottleneck between the corporate network and the Internet. They were also becoming the source of many outages, compromising our security by leaving certain applications unavailable. This paper is a case study of the issues we were facing, the thought processes used to address them, and the final outcome of the chosen solution.

Situation

Physical Layout

Our Internet firewall architecture was probably typical of most mid-sized companies. Figure 1 is a rough diagram of the architecture. It involved three firewalls, although only two are shown in the diagram, with some load balancers. The proxy-based firewalls that we were using are very expensive in terms of hardware and software license. The load balancing solution was a relatively inexpensive way we could utilize the expensive multiple copies of proxy-based firewall licenses we had acquired during the mergers and acquisitions.

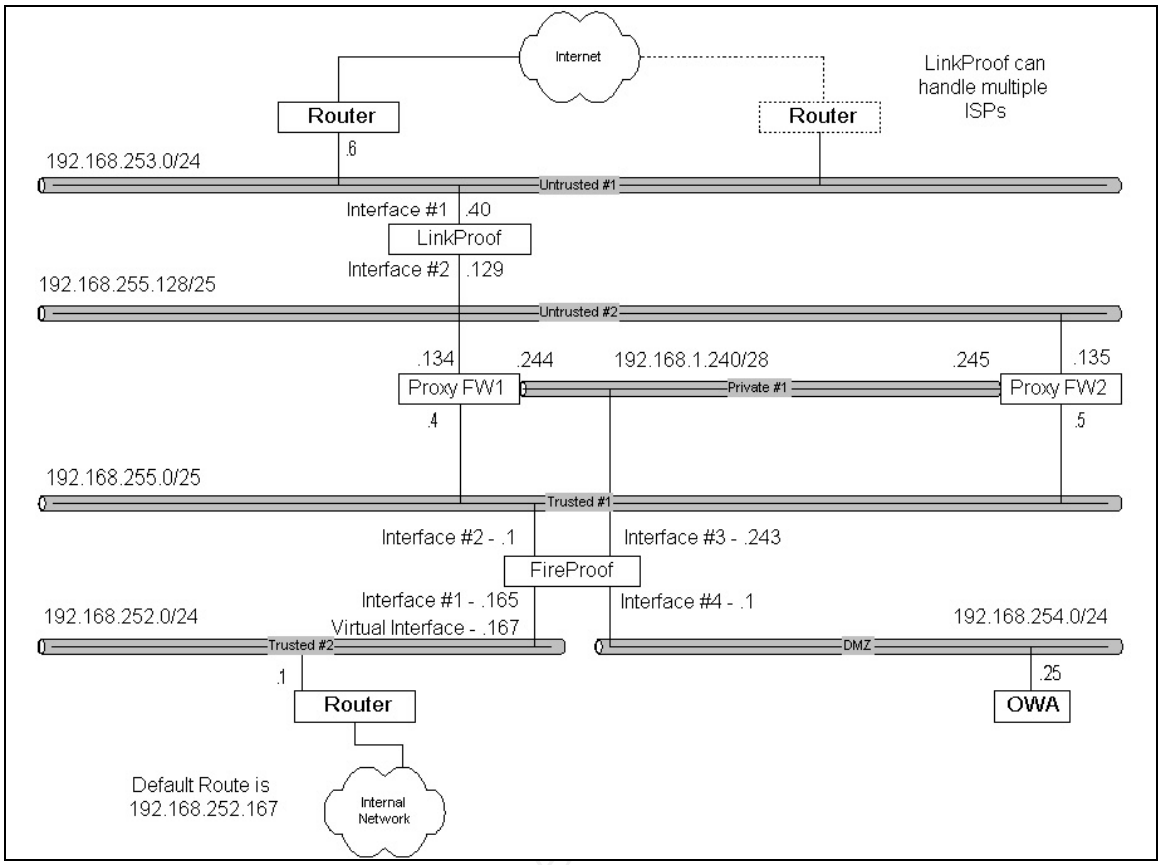

Figure 1

Load Balancers

The load balancers allowed us to add firewalls to handle additional load. The external load balancer (linkproof) distributed the load of incoming traffic from the Internet among the firewalls. The internal load balancer (fireproof) distributed the load initiated from two networks. It distributed load from the internal network across the firewalls, whether it was destined to the DMZ or to the Internet. It also distributed traffic originating from the DMZ across the firewalls, whether it was destined for the internal network or to the Internet. This was possible because the fireproof is a four-port device capable of being split logically into two devices. The load balancers helped us address some of the initial load issues but as you will see they did not allow us to keep up with the pace of growth. The application level gateways were not within the limit of our budget constraints. Also there were issues that it could not address, firewall reliability being one of them.

Firewalls

The firewalls were proxy-based firewalls, meaning they looked into the packets of each connection to make sure that the connection was following the protocol of the application being used.[1] They also allowed for the configuration of rules which would limit connectivity based on source address, destination address, source port, destination port, which interface the connection came in on, and which interface the connection was going to leave on. The firewalls also had general proxies in the event that there was no specific proxy for a particular application.

Availability

The decision to use application level proxies was one of factors that had put the company in the situation of redesign. Designing an Internet firewall architecture around application level gateways is a good choice if they support all of the applications that you plan to run. The company had deployed an application to its Internet user base that was based on UDP as well as TCP. We have had the policy that all Internet-accessible applications would live in the DMZ. While the traffic from the client to this application was TCP-based, the backend communication was UDP. To make this application work we created a general UDP proxy for the backend communication between the DMZ and the Internal network. This application worked but we ran into reliability issues with the firewall passing the backend traffic. The general UDP proxy would quit passing the traffic. We worked to try to fix the problem for several months without success. Finally we decided that our application level gateway was the wrong solution for the backend firewall.

Bandwidth Increase

With mergers and acquisitions came an increase in Internet usage. The mergers and acquisitions caused an influx of new Internet-based applications on the network. Also the number of employees doubled, which increased the amount of Internet surfing. The company's Internet usage went from the bandwidth of a T1 (1.544Mbps) to about 18 Mbps over the course of three years. This was a huge increase in demand that the existing Internet firewall architecture struggled to satisfy.

Mission Critical

Internet connectivity came to viewed as mission critical. Several groups were demanding availability of the Internet including the law department, the messaging group, the e-commerce group, as well as others. This would generally not be a problem but the groups were not prepared to fund the security

[1] Curtin, Matt. Ranum, Marcus. "Internet Firewalls: Frequently Asked Questions" Dec. 1, 2000 URL: http://www.interhack.net/pubs/fwfaq/ (January 14, 2002)

aspect of Internet connectivity. So firewall funding became a huge factor in the need for a redesign because of the demand for availability.

Default Route

Many other factors influenced the redesign of the Internet firewall architecture; but I will only mention one more. Even though we were using proxy-based firewalls, we implemented them as transparent proxies. This would keep us from having to configure all of the browsers of the users. For this to work a default route had to be distributed within the network. This would funnel all traffic not destined to an internal network to the Internet firewalls. This proved to be an issue with many internal boxes, some misconfigured, bombarding the firewalls with disallowed traffic. We enlisted the help of the network group to install an access list on the router feeding the Internet firewalls. We tried many times to apply an access list but every time we broke Internet access altogether.

Summary

There were many factors that influenced the need for a redesign of the Internet firewall architecture.

The main issues were:

- UDP reliability issues.
- An increase in demand.
- Internet connectivity being viewed as mission critical.
- Illegitimate traffic bombarding the firewalls because of the default route.

All of these issues along with the decline in spending as a result of the events of September 11, 2001, let to the need for a redesign of the Internet firewall architecture.

Available Options

In the design of an Internet firewall architecture there are many options available. You have the choice of different firewalls, including application level gateways, circuit level gateways, hybrid firewalls, stateful inspection firewalls, or packet filtering firewalls. A combination of multiple types of firewalls could be used. You could choose to load balance across multiple firewalls or buy one big firewall that could handle the entire load. In the area of load balancing you could choose to do it in software or hardware. Inbound traffic and outbound traffic could be segregated into an independent inbound path and an independent outbound path.

Firewalls

In the area of firewalls we have four major options available. In the following paragraphs we will explore some of the pros and cons to each firewall type. While I will not go into great detail about the differences I will highlight the main issues involved in the decision of which type of firewalls to use in the resulting recommendation of a new Internet firewall architecture.

Application Level Gateways

Application level gateways are credited with providing the most security. This level of security comes at a cost. Because application level gateways inspect the packets at the application layer of the TCP/IP layered model, they have the most overhead when it comes to performance.[2] Application level gateways are generally more expensive compared to the other firewall architectures. This is understandable given the time needed to develop the application level proxies for each protected application. Another problem with application level gateways is they do not always support applications that a business wants to run. Application level gateways provide the best security for applications that they are aware of, but this is not always what the business wants run.

Circuit Level Gateways

Circuit level gateways provide a little less security than an application level gateway but you gain a little bit of performance boost from this. Another benefit is that, like an application level gateway, the circuit level gateway doesn't allow a direct connection between the client and server. The connections are terminated and started at the gateway.

Hybrid

Hybrid firewalls are another firewall technology that blurs the lines between the other technologies. Hybrid firewalls are typically made up of combinations of the other technologies. Vendors that have historically provided application level gateways are implementing low level filtering to help improve performance. Also vendors that have historically provided packet filters or stateful inspection firewalls are adding features that resemble application level proxies to offset some of the weaknesses of their products.[3]

---

[2] Wack, John. Cutler, Ken. Pole, Jamie. "Guidelines on firewalls and firewall policy – Recommendations of the National Institute of Standards and Technology" January 2002. URL: http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf (January 6, 2003)

[3] Wack, John. Cutler, Ken. Pole, Jamie. "Guidelines on firewalls and firewall policy – Recommendations of the National Institute of Standards and Technology" January 2002. URL: http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf (January 6, 2003)

Stateful Inspection

Stateful Inspection firewalls are a relatively newcomer to firewall architecture. Stateful inspection firewalls were developed by building additional functionality on top of the packet filter model. While the technology is less secure than an application or circuit level gateway it is more secure than a packet filter because it keeps a state table of active connections. This makes the firewall easier to maintain, as you don't have to define rules for replies to legitimate traffic. It also makes it more secure because rules use for replies could also be used to exploit vulnerabilities in internal boxes.

Packet Filter

Least secure of firewall technologies are packet filters. Packet filters decide whether to allow a packet strictly on that packet alone. While it inspects information in the TCP header it doesn't keep a state table of connections. It only looks in the TCP header for the source and destination port numbers to compare against its rule set. Given that it does the least amount of checking it is generally the fastest of the firewall technologies.

Load Balancing

In the area of load balancing two main technologies exist. The next couple of paragraphs will explore the two technologies and their pros and cons. While again I will not go into depth on each technology I will discuss the main issues that were considered.

Software-based

The first load balancing technology we will look at is software-based load balancing. Software-based load balancing has the advantage of not needing additional hardware, as generally it will run on the firewall itself. It will allow you to cluster multiple firewalls together such that the load gets distributed across all of the firewalls. It has a disadvantage of adding additional load on the firewalls themselves. Also depending on the number of firewalls you plan to use, software-based load balancing can be more or less expensive than hardware-based load balancing.

Hardware-based

The other load balancing technology evaluated was hardware-based. Hardware-based load balancing has the disadvantage of needing additional devices but has the advantage of not adding load to the firewalls themselves. Hardware-based load balancing will allow you to put multiple firewalls between them and they will distribute the load among the firewalls. This has the potential to offset the

additional hardware cost by off-loading processor cycles from the firewalls themselves making the solutions need less firewalls horsepower.

The New Solution

Load Balancing

Armed with this general knowledge of the technologies, it was time to start making some decisions. Given that the company already had an investment in hardware load balancers the choice was easy to continue in that direction. The company had already invested more than $20,000 in a hardware-based load balancing technology.

Firewall Types

Deciding what firewall architecture to move forward with was a much more difficult decision. The firewall technology used would be directly related to the level of security the corporation would have from hackers. Moving forward, we decided to use a combination of application level gateways and stateful inspection type firewalls. This would retain our investment in the application level gateways but allow us to gain some of the needed benefits of stateful inspection type firewalls. We were in need of speed and reliability improvements; and stateful inspection type firewalls could provide both.

Physical Layout

Once we decided to move forward with two distinct firewall technologies there was an issue about how to marry the two technologies. One of the possible options would be to put the application level gateways as the external firewalls protecting devices on the DMZ as well as internal networks. This option put the stateful inspection firewalls between the DMZ and the internal networks. This design made sense because the traffic going between the DMZ and the internal network did not utilize the strengths of the application level gateway. There were no proxies for most of the traffic passing between the DMZ host and the Internal host. If we were getting no benefit from having the application level gateways then we should use another technology. Stateful inspection type firewall would still provide good security while giving the additional benefit of increase performance. Figure 2 shows this option.

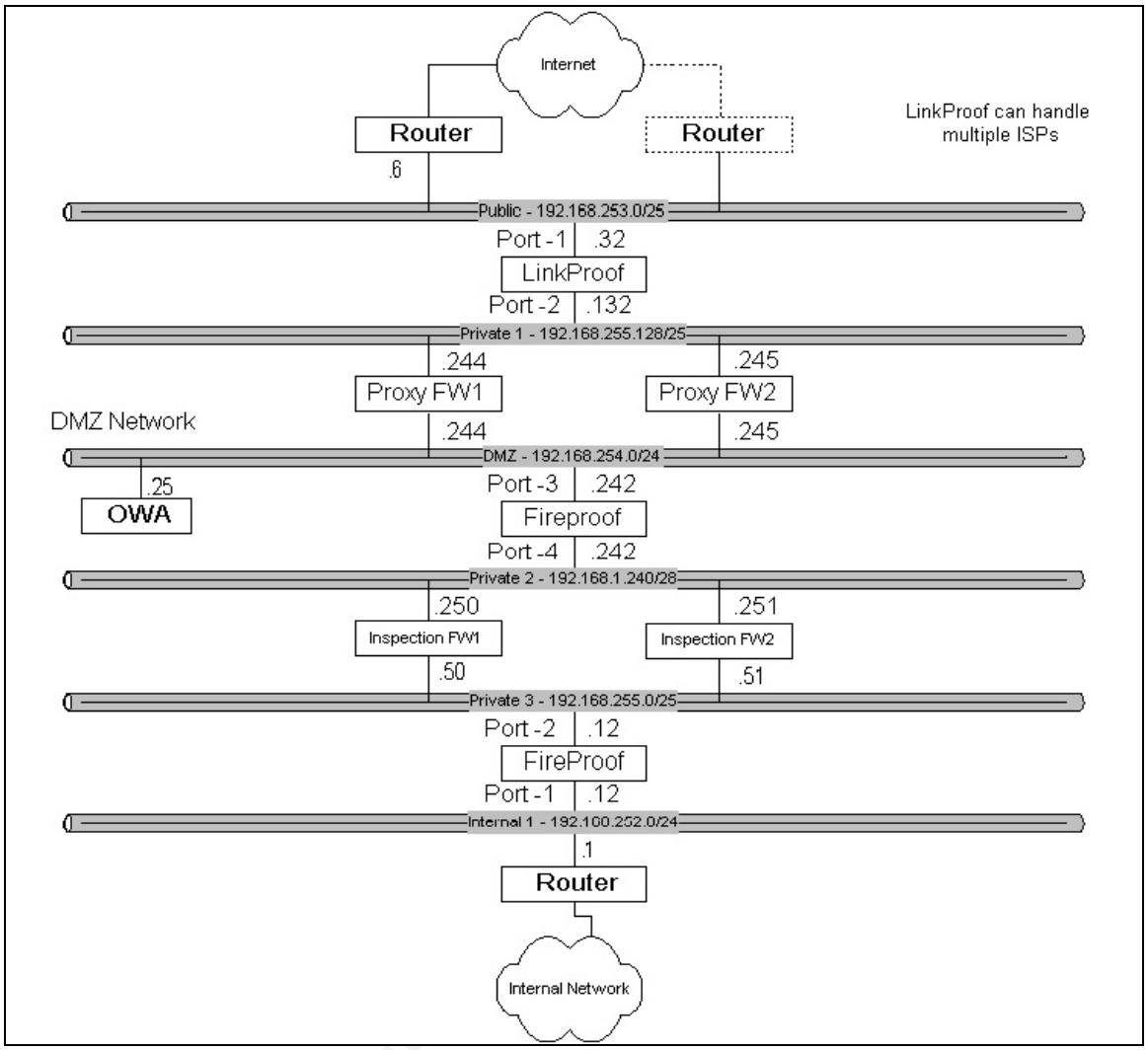

Figure 2

This option would solve a couple of problems. It would utilize the faster stateful inspection firewalls to block all of the illegitimate traffic from internal users, thus preventing wasted firewall cycles. It would also get around the issue of having to use general proxies between the DMZ and internal network. This is an area in which we were having reliability issues with UDP traffic. This solution still required all Internet surfing to pass through the proxy firewalls, which was one of our biggest problems. Budget constraints would keep us from being able to add additional application level gateways, and while this solution might handle the load initially we want something that would scale a little better.

The next design shown in figure 3 was another way to marry the application level gateways with the stateful inspection firewalls. This design would split the inbound and outbound paths. The inbound path would be the design shown in figure 2, where the application level gateways were the external firewalls and the stateful inspection firewalls would be utilized between the DMZ and internal

network. This would give the company the maximum protection for inbound connections while providing a backend firewall that was more in tune with our needs. The outbound path would utilize the stateful inspection type firewall to provide connectivity to the Internet for internal users. From a security standpoint this would be acceptable because the company was already using other products (CISCO's 590 Content Engine in conjunction with Websense) to restrict access to certain types of web content on the Internet. Looking at figure 3 the outbound path is on the left and the inbound path is on the right. In this design the load balancers and routers are shared between the two paths.

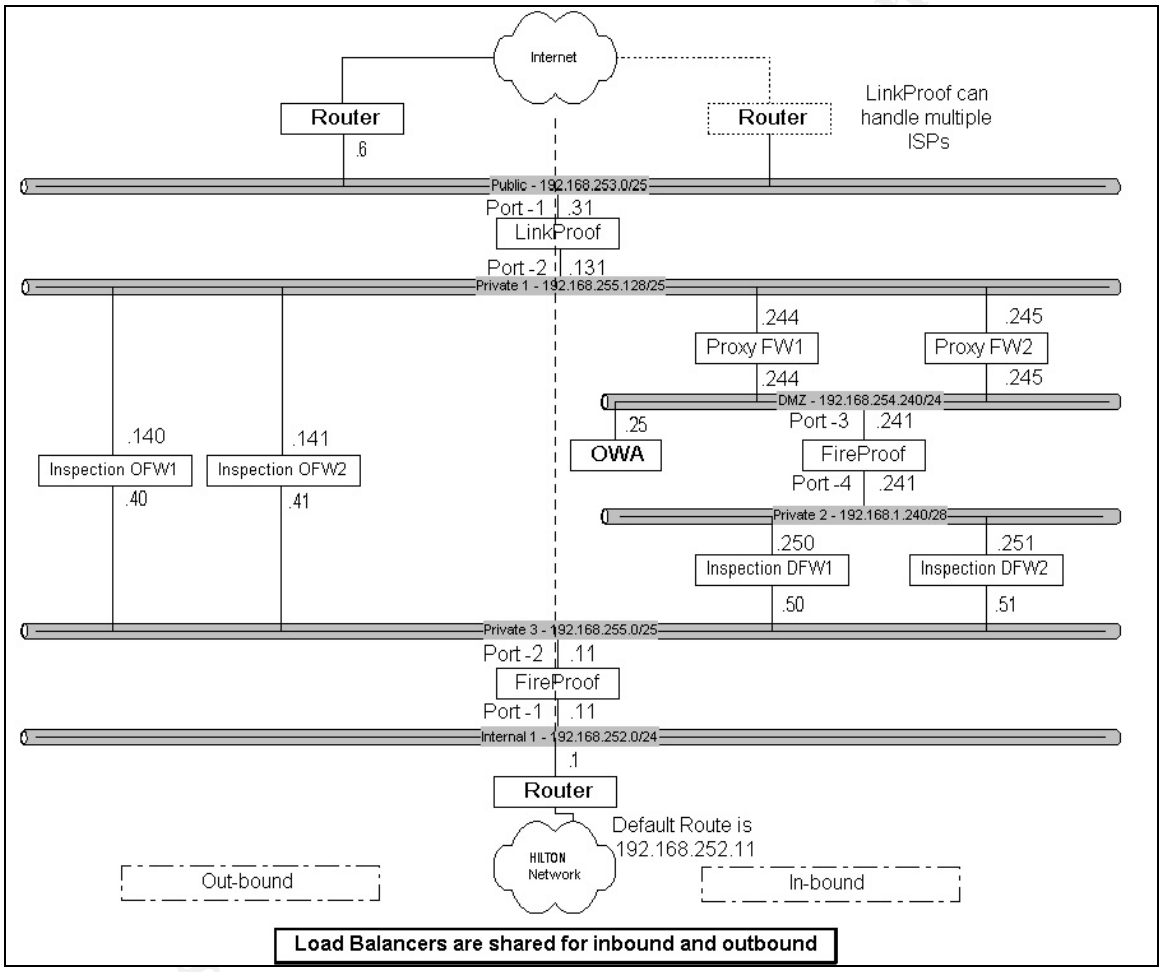

Figure 3

This option is a little more complex but it would solve the issues we were facing without compromising security. This option actually enhances security by diversifying our firewall technologies and providing a higher level of availability. It would also keep us from throwing away our current investment in firewalls and load balancers. Since the security group was willing to take on the increased complexity of this option, to solve the issues we were having with our Internet architecture, this would become the new physical layout of the Internet firewall architecture.

Stateful Inspection Firewall Selection

Now that we had decided on a physical layout for the new Internet firewall architecture is was time to choose what stateful inspection firewall we were going to use. There were a few key criteria in making this decision. To minimize the complexity of support we wanted to choose a vendor that we currently used and had in-house expertise in. The other main criterion was cost. We have very little money to invest in the Internet firewall architecture, which is why we were in this situation. If money wasn't a factor, then bigger faster application level gateways could have been purchased, preventing this whole process. So with the two main criteria being cost and supportability, we set out to pick a stateful inspection firewall technology.

Vendor Selection

Evaluating which vendors we had that offered stateful inspection firewalls, we only found a few choices. Our network is primarily composed of CISCO gear, so this made the PIX a strong contender. We didn't have any production PIX in our network today but with the backing of the vendor we could probably get up to speed quickly. We also had a few Netscreen firewalls distributed throughout various points in the network. While the Netscreen is a capable firewall it doesn't offer some of the features other stateful inspection firewalls do and we were looking at eliminating this vendor. Another candidate was Linux. Since Linux is a prime platform for security tools our security group had much experience with the OS. With the introduction of the netfilter code and the 2.4 stable release of the kernel, which included this code, Linux had been given the capability of being a stateful inspection class firewall.[4] In the end we determined that the CISCO PIX and the Linux 2.4 kernel would move forward to the stage of comparing price.

Before we could accurately compare pricing we needed to understand what kind of box would give us adequate bandwidth from Linux. After some research we only found a little information about its capability. Considering an average frame size of 512 Bytes it appeared that a moderately sized Intel-based PC could push upwards of 80Mbps of traffic through a synchronous rule set of 200 rules.[5] Other resources on the web indicated that the majority of problems with the netfilter code in stateful inspection mode was with running out of memory for the connection state table, /proc/net/ip_conntrack.[6] Using the specifications from the box used in the test as a base line we decided to add additional memory so we could adjust the connection state table to handle our client base. Also we knew

[4] Bandel, David A. "Taming the Wild Netfilter" September 1, 2001
URL: http://www.linuxjournal.com/article.php?sid=4815 (September 2001)
[5] URL: http://www.hipac.org/test/results.htm (November 2002)
[6] Stephens, James C. "Connection tracking" April 5, 2001 URL:
http://kalamazoolinux.org/presentations/20010417/conntrack.html (December 2002)

we would purchase two boxes to balance the load with the idea that one could handle the load in the event of a failure on the other. We would need a total of four servers, two for outbound firewalls, and two for firewalls to reside between the DMZ and the Internal network.

Price Comparison

Given that we were on a very tight budget pricing was very critical. Four IBM X330s with the PIII 1.4GHz processors, upgrading them to 1GB of RAM and using a ServeRAID controller to mirror the two 18.2GB 10K rpm Ultra160 drives came in at roughly $16,000. Four CISCO PIX 515s were going to cost roughly $32,000. Performance wise, the CISCO PIX 515 could probably outperform the IBM server, but with the difference in price we could afford eight IBM servers. With the difference in price and the ultra tight budget we decided to give Linux a chance at making a robust stateful inspection-type firewall.

Recommendation

Physical Layout

To eliminate the issues we were having, the security group decided to redesign the Internet firewall architecture. The redesign would have a physical layout of that shown in Figure 3, where the inbound and outbound Internet paths would be separated. Linux-based stateful inspection firewalls would be used in two spots: As the outbound firewalls and the firewalls between the DMZ and the internal network. The security group thought that this design would substantially increase the robustness of the Internet firewall architecture.

Stateful Inspection Firewalls

The use of Linux-based stateful inspection firewalls, as outbound firewalls, would help two of the main issues we were facing. They would handle the volume of traffic the company was utilizing to the Internet. They would also effectively block the illegitimate traffic some of our Windows PC's were generating, mainly NetBIOS broadcast. Using stateful inspection firewalls rather than application level gateways would enhance availability of resources, therefore increasing our security.

Using Linux-based stateful inspection firewalls as firewalls between the DMZ and the internal network made perfect sense. The company was not utilizing the strengths of an application level gateway at this security point, as most of the traffic between the DMZ and internal network was not supported by application specific proxies. In fact, the general UDP proxy provided by the application level

gateway vendor was frequently breaking, causing an outage of all UDP traffic between the DMZ and the internal network. There were no valid reasons to not use a stateful inspection-type box.

Load Balancing

With the savings from going with Linux-based stateful inspection firewalls, it was recommended that the company purchase redundant load balancers. This would eliminate a single point of failure for the Internet firewall architecture. The final recommended design can be seen in figure 4.

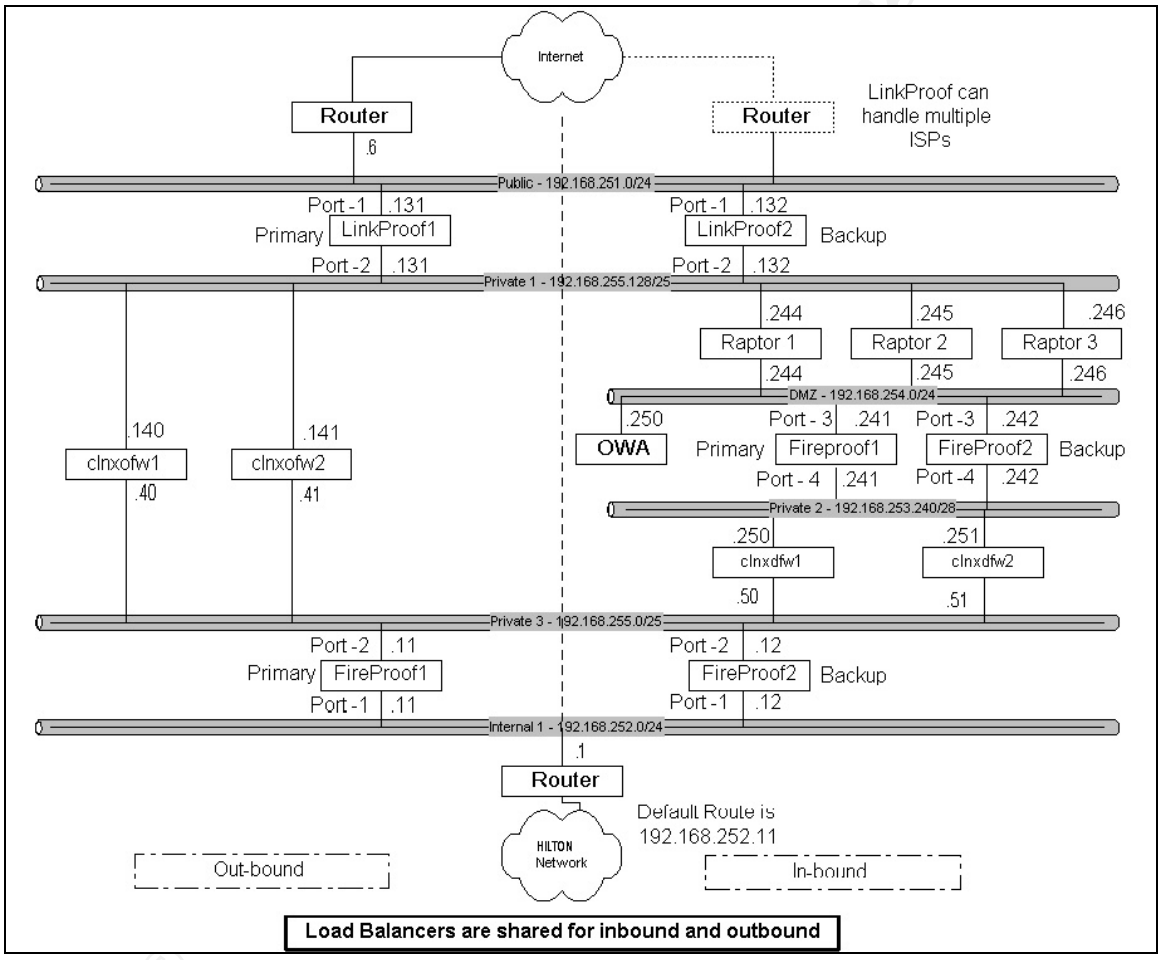

Figure 4

Results

Performance

After implementing the Internet firewall architecture shown in figure 4 we have seen some interesting results. The company's Internet bandwidth peaked upward to 20Mbps within a week. During the initial push into production only Raptor 3 was handling the load at that point in the solution and it was running 70% to 90% idle. The majority of our load is internal people surfing. The two outbound firewalls generally run 90% idle during peak. In fact, no box in the solutions is less than 75% idle at any given time.

One Concern

One concern with using Linux was that the connection state table had been known to fill up. With 1 GB of memory in those boxes netfilter set the connection state table max to 65,535. During the peak of a day there are generally less than 20,000 entries in the state table of each outbound firewall.

Resolved Issues

The implementation of the new Internet firewall architecture has been a success. We have plenty of available resources to handle outbound surfing in addition to denying illegitimate internal traffic bombarding the default route. No UDP issues from the DMZ to the internal network have been experienced since the new architecture was put into place. Availability of our Internet access has increased, as no firewall issues have caused an Internet outage. While we continue to monitor the environment we feel we have successfully improved our Internet firewall architecture.

List of References

1. Bandel, David A. "Taming the Wild Netfilter" September 1, 2001. URL: http://www.linuxjournal.com/article.php?sid=4815 (September 2001)

2. Curtin, Matt. Ranum, Marcus. "Internet Firewalls: Frequently Asked Questions" Dec. 1, 2000. URL: http://www.interhack.net/pubs/fwfaq/ (January 14, 2002)

3. Stephens, James C. "Connection tracking" April 5, 2001 URL: http://kalamazoolinux.org/presentations/20010417/conntrack.html (December 2002)

4. Wack, John. Cutler, Ken. Pole, Jamie. “Guidelines on firewalls and firewall policy – Recommendations of the National Institute of Standards and Technology” January 2002. URL:http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf (January 6, 2003)

5. URL: http://www.hipac.org/test/results.htm (November 2002)

Last Updated: July 27th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event