Global Information Assurance Certification Paper

Managing Network Firewalls –A Love/Hate Relationship

James P. Medeiros Jr.
SANS GSEC Practical
April 25, 2002
Version 1.4 Option 1

Introduction

In today's rapidly changing network environment, one thing is certain. If you stand still for even a moment you **will** be left behind. Constant upgrades to hardware and software leave us spinning with the never-ending task of evaluating our network configurations. Constant tweaking is required to adapt to the constant changes of new and existing network applications.

One of the major systems which require constant attention is a network Firewall. For any application to communicate across present day networks they will eventually have to cross a Network Firewall. These Gatekeepers are a double edged sword. On one hand they are an immeasurable tool for, implementing a security policy. On the other hand, they are high maintenance considering the constant changes in network configurations and applications flowing through the network infrastructure. We will discuss some of the problems and possible solutions that you may encounter in the day-to-day administration of a Network Firewall. We will also touch on ways to troubleshoot applications, which are flowing through your Network Firewalls.

History

First we must step back and take a look at how this all started. Back in the dark ages, say about 1969, when a research group composed of both the military and academic communities gave birth to the Advanced Research Projects Agency Network (ARPANET). The initial four sites (UCLA, Network Measurement Center SRI, Network Information Center UCSB and Culler-Fried interactive mathematics UTAH) were to become the first network. **(1)**

It wasn't until 1988 that an event took place that would forever change the mindset on security within the Internet. A message hit the TCP/IP Internet mailing list stating: "We are currently under attack from an internet VIRUS!" Peter Yee of the NASA Ames Research Center reported the incident. This virus, later to be called the "Morris Worm" affected Berkley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames. As a result of this situation, several Internet mailing lists were created dedicated to sharing information in order to develop procedures to prevent further disruptions. **(2)**

In the 1980's the first firewalls started to appear. Cheswick and Bellovin, in a definitive text on Internet Firewalls, state "an Internet firewall has the following properties: it is a single point between two or more networks where all traffic must pass (choke point); traffic can be controlled by and may be authenticated through the device, and all traffic is logged."**(3)** Initially routers were used to separate a network into smaller LANs. An example might be to separate the purchasing department from the research department. The intent here may be to prevent network problems from spilling over to

the whole company. The first security firewalls were used in the early 1990's. They were basically routers with filtering rules. These firewalls were limited and soon replaced by more elaborate and adjustable firewalls called bastion hosts. **(2)**

Firewall Types

Modern Firewalls come in three basic types:

- Packet Filters
- Circuit gateways
- Application Gateways

Packet Filters

Firewalls of the packet filtering variety are the most basic of the three. They perform packet inspection looking at the IP address, port and protocol ignoring everything else. A packet filter can be configured to allow or disallow packets based on IP address, port, or protocol. No attention is paid to the application. A packet filter can be very effective and is capable of blocking all traffic. They are not foolproof however and can be tricked into passing traffic by falsifying information such as source IP address (Spoofing). Since the payload is not inspected it could easily contain malicious code. Installation and configuration are simplest with packet filters which are its strengths. **(2)**

Circuit Gateways

A Circuit Level Gateway or Circuit Relay is a middle tier Firewall. This type of Firewall goes a step further in that it can validate the connection. By using configurable rules, both ends of the connection can be validated. This validation can be based upon the following:

Source and or destination IP Address
Source and/or Destination Port
Protocol
Time of Day
User and or password

Traffic is not allowed except through valid sessions. This can make IP spoofing more difficult and makes up for the lack of source IP address validation in the UDP protocol. A disadvantage of the Circuit Level Gateway is that it works at the transport layer which may require some program modifications of the transport function (Winsock).

Application Gateway

An Application Gateway or Proxy Firewall works at the highest tier. As its name implies, this type of Firewall functions at the Application Layer. A much more complex system of rules can be configured in this type of Firewall. Working as a proxy, the

firewall establishes sessions and performs data exchange on behalf of the systems behind it. Very extensive logging capabilities make this type of Firewall very useful for intrusion detection and network monitoring. Considered the most secure of the three Firewall types, the configuration and management are much more complex. Most often implemented as a stand alone appliance, a large amount of processing power is required. Many of the newer Firewalls would be considered a Hybrid of more than one type of Firewall. They can act as a lower tier Firewall normally stepping up to a higher tier only when required by the policies set in their configuration. This makes them much more efficient. Hybrid Firewalls are not new however; they have been around since the beginning. **(2)**

Firewall Implementation

Even in the short time that I have been involved in this field I have noticed a swing in the emphasis of firewall implementation. When Firewalls were first introduced they were the ultimate solution to all of our network security woes. We started installing them at our Local Area Network gateways and our blood pressures plummeted immediately as we sighed a big breath of relief.

We installed these firewalls in specific locations where our protected network met with the outside world and all of its vulnerabilities. We then spent some time wading through the documentation scratching our heads as we poked hole after hole through the Firewall to get all of those nuisance applications and their protocols to talk with each other. We soon found it easier to poke a hole in both directions and with a span of ports rather than taking the time to figure out which way we really needed the traffic to flow and with what ports. It was soon discovered that this thing called Dynamic Host Configuration Protocol (DHCP) was creating havoc with our rules. So, what is DHCP?

DHCP Defined

DHCP was created by the Dynamic Host Configuration Working Group of the Internet Engineering Task Force. IETF is a volunteer organization which helps to define protocols used on the Internet. It was created to ease the burden of large network administrators by automatically providing common configurations to client systems when they are connected to the network. DHCP uses UDP ports 67 and 68. **(4)**

According to RFC 2131 **(5)** a client wishing to obtain a new configuration on a network would use the following basic steps.

1. The client broadcasts a DHCPDISCOVER message on its local physical subnet.
    - The client may suggest values for its configuration in this message.
2. One or more servers may respond with a DHCPOFFER message that includes an available network address and possibly other configuration parameters as options.

- The servers should check that the offered address is not already in use and may do this with an ICMP Echo Request.

3. The client receives one or more DHCPOFFER messages from one or more DHCP servers.

    - If the client receives multiple responses it will have to choose which offer to accept.

4. The servers receive the DHCPREQUEST broadcast from the client.

    - The client must specify which server it has accepted the offer from in its request.

5. The server which has been accepted will respond with a DHCPACK.

    - If the selected server is unable to satisfy the DHCPREQUEST message, the server should respond with a DHCPNAK message.

6. The client receives the DHCPACK message with configuration parameters.

    - The client notes the duration of the lease and should perform a final check on the parameters.
    - At this point the client is configured.
    - In the case of a client receiving a DHCPNAK message, the client restarts the configuration process again.

7. The client may choose to release its lease on a network address by sending a DHCPRELEASE message to the server.

The following are the steps that may take place when a client already has an address and wishes reuse it.

1. The client broadcasts a DHCPREQUEST message on its local subnet.
    - The message includes the client's network address.

2. Servers with knowledge of the client's configuration parameters respond with a DHCPACK message to the client.

    - The server should respond with a DHCPNAK if the request is invalid.

3. The client receives the DHCPACK message with configuration parameters.

    - The client performs a final check of the parameters and notes the duration of the lease.

4. The client may choose to relinquish its lease on a network.

The security ramifications of installing a DHCP Service are as follows. In its basic configuration, any computer physically connected to the local network (including rogue computers) can make a request and will receive the necessary configuration to participate on the network.

Some of the steps which may reduce the risk are to configure the DHCP Server with only enough addresses to satisfy existing clients. The problem with this is that as leases expire, an opportunity arises for the address to be taken by an unauthorized host. Also, by constantly adjusting the available address allocation, you will increase the time spent on administration which in term negates the purpose of DHCP in the first place.

Another option may be to reserve addresses on the DHCP Server based on a MAC address. The Media Access Control address is unique to each node or computers (more specifically the network card) on the network. This will make it more difficult for an unauthorized host to gain access but again not impossible (MAC Address cloning). Again the point here is to add several layers of defense an reduce the threat (Layered Defense).

How does DHCP affect our Network Firewall Administration? Well most firewall rules revolve around allowing or not allowing specific IP addresses into or out of our network. If the IP addresses are a moving target, it will be difficult to maintain effective rules. This normally affects the inside addresses though, so many Firewall administrators resolve this by modifying the rules to allow entire subnets through. Now, instead of allowing application "X" into our network to talk to client "Y" on TCP port 23 (telnet), we allow application "X" the opportunity to communicate via telnet to all of the addresses within the subnet. What are the possible solutions to this? Well one obvious solution would be to give client "Y" a static address. Another way would, like the example above assign client "Y" a reserved address based on its MAC address by setting up a table on the DHCP Server..

Now, let's take a look at another Network Firewall security paradox, Network Address Translation (NAT). There is a growing trend to use NAT on Network Firewalls as a security feature to hide internal network addresses. If all of the traffic passing through the firewall were to appear to come from the same address, say that of the Firewall itself, it might make it difficult to know the internal network IP address scheme. This may deter an outsider from using exploits based on IP addresses to establish a trust and gain access. This would seem to increase our security but as you will see, at a price.

History of Network Address Translation:

NAT was first introduced in RFC 1631 **(6)** as a short term solution to the depletion of available addresses on the Internet. By using a NAT device at the border of a Network domain you could maintain any address scheme within the local domain without being concerned with the effect on their domains since thy will never interact.

These addresses would be converted by the NAT device to legal globally assigned unique addresses when communication with outside networks was necessary.

Several newer RFC's have been written to further define Network Address Translation. In RFC 3022 Basic NAT is defined as "A method by which IP addresses are mapped from one group to another, transparent to end users". This RFC introduces or extends basic NAT to include port translation or NAPT (Network Address Port Translation). With NAPT network addresses and their TCP/UDP (Transmission Control Protocol/User Datagram Protocol) ports are translated into a single network address and its TCP/UDP ports. By using both operations, you are able to connect several private network addresses to and external public network with globally unique registered addresses. **(7)**

Another benefit of using NAT is that since the inside network addresses are never exposed to the public domain, if your network changes providers, or the corporate network reorganizes its IP scheme, only the NAT device needs to be modified.(6)

However, there are several problems with using NAT. Some applications, FTP for example, may use a bundled session approach (Control and Data sessions) to accomplish data transfer. In the case of ftp, by default, the data session is established from the server's source port 20 to the client's port used to establish the control session. In this case an Application Layer Gateway (ALG) is required to monitor the control session and modify the private address and port number with the externally valid address and port number. **(8)**

A major difficulty with using NAT within a firewall is with IPSEC. IPSEC is used in many Virtual Private Networks (VPN's). As its name implies, it provides secure communication on the Internet. IPSEC has two modes, Transport, and Tunnel. Transport mode applies IPSEC protocols to the IP packet and leaves the original IP headers visible. Transport mode can be used only in host to host IPSEC VPN's. With Tunnel mode IPSEC, the original IP packet is encapsulated into an IPSEC packet with new IP headers. This effectively hides the original IP packet from view. Tunnel mode must be used in host to gateway communication. Since NAT alters the IP header information, it breaks most IPSEC solutions. When IPSEC is going to traverse a NATed network, extra care must be taken with the placement of the VPN as well as configuration. **(9)** RFC 2709 "Security Model with Tunnel-mode IPSEC for NAT Domains" contains solutions for deploying IPSEC across a NATed Domain. **(10)**

Unfortunately NAT can make our job of troubleshooting and tracing data flow a little more complicated. Lets take a look at a scenario where a company has several geographically separated facilities.. As a network administrator at our fictitious company's Houston Site, I find it difficult to accept the concept of allowing traffic from another site or company to enter my network with just one address, when I know that it is the address of that sites Firewall, and that I have no way of guaranteeing that the outside company has not been infiltrated. I could now vulnerable to exploitation. So I, in the interest of security, created a policy, which forbids the introduction of NATed addresses

into my Network. You, an administrator in Boston, try to share with me the error in my ways. You tell me that by allowing all of my internal addresses through the Firewall, I am making it easier for the bad guys to map out the network and narrow down the possible addresses and possibly OS's to exploit. You also explain that there is no way that you are going to turn off NAT so that I can get that warm fuzzy seeing all of your addresses as they pass through my network. Again, if we just work together and share our network security policies we can put each other at ease and work together in troubleshooting network difficulties in an Address NATed environment (Why can't we just all get along).

Then, there are the stories of how smart hackers are. A firewall is just a nuisance, like locking your car door to a thief. All the thief has to do is use a "hole puncher" on one of those cool side windows and he/she is in. Of course, now, they have to try to start the car without a key. Or maybe you had a car alarm and the "CLUB". I think this analogy works well with our Network Security as well. The car alarm sounds a lot like an automated Intrusion Detection System. How about the ignition key? That might be the password authentication on our network. And let's not forget the club. Maybe that's our Host based firewall protecting the high risk assets in our network. Why stop there. This could go on with the tinted windows or the cover we put over our cargo compartment in that brand new SUV to hide our goodies from the would-be thief. What do you think that would be on our network? How about Network Address Translation (NAT)? I just read an article which took this analogy a step further. Jon Lasser writes that "If your neighbor leaves his keys in the steering column, he's not necessarily the only person who loses out. A drunken joy rider could go on a tear through the neighborhood, ripping through the shrubbery and knocking over telephone poles". He goes on to say that: "Similarly, intruders turn easily-cracked, poorly-administered sites into launch points for further attacks." **(11)**

Troubleshooting Firewalls:

Everybody blames the Firewall. One thing that I have learned is that whenever there is a problem with a network application, it is always the Firewall that gets blamed. How do we determine where the problem lies? Nowadays we are putting these Firewalls everywhere. On our host machines as well as our domain borders. Say we have a company with several geographical locations. We have a financial application at our Houston location and it needs to communicate with a Server located in Boston. The client terminal keeps getting a generic error that states that the host cannot be reached. Where does the problem lie? How can we troubleshoot this? Well, we need a program, which will monitor the traffic at various points along the circuit with the capability of filtering on something specific about this applications traffic. So, the user of the client machine calls up Boston's IT department and says I can't reach the server and of course accuses Boston's Firewall of blocking the traffic. The administrator at Boston wants to verify that his Firewall is not to blame and uses a program called TCPdump to check the traffic through his Firewall.

What is TCPdump?

TCPdump is an tool which can be used to analyze network traffic. Once installed, TCPdump must be run with root privileges. Its output can be intimidating at first but if you invest a little time studying its format it becomes clear. Many of the commercial analyzers use a similar format as well. TCPdump comes complete with a filter language which will allow great flexibility in its output. It is available in most UNIX flavors as well as Windows (Windump). You can download TCPdump from:**(12)** ftp.ee.lbl.gov/tcpdump.tar.Z.

You will also need libpcap, which implements a portable framework for capturing low-level network traffic. You can find it at: ftp://ftp.ee.lbl.gov/libpcap.tar.Z.

The Windows variant of TCPdump is available at: http://windump.polito.it/install/bin/WinDump.exe.

You will also need WinPcap in order to run Windump. You can find WinPcap at: http://winpcap.polito.it/install/bin/WinPcap_2_3.exe

Now back to our Boston problem. Realizing that the problem may not be a Firewall at all. It could be the circuits, the server itself, the client machine, or any Firewall or Router along the path. We need to start somewhere and the Administrator at Boston has already eliminated things like the circuit and Router for the moment because he has not received any other complaints, which would lead him to circuit troubles. So he needs to take a look at the inside and outside interface of the Firewall. By using TCPdump he can filter on something specific about the traffic in question. For example he could filter on the source or destination port, the source or destination address or any combination there-of. He decides to use the destination IP address because he knows that is a constant and will not be affected by NAT along the way. He starts two terminal sessions on the Firewall and types the following commands.

Terminal 1: **Tcpdump –ni eb0 host (Boston Server IP)**
Terminal 2: **Tcpdump –ni eb1 host (Boston Server IP)**

Figure 1

Outside Interface:

```
Boston FW1 %  tcpdump -vni eb1 host (Boston Server IP) and port 23
tcpdump: listening on eb1
07:49:00.796125 (Houston Clt IP) > (Boston Svr IP).23: S 28011941:28011941(0) win 8192 <mss 1460> (DF) (ttl 128, id
47961)
07:49:00.796212 (Boston SVR IP).23 > (Houston Clt IP).3974: S 618675114:618675114(0) ack 28011942 win 33580
<mss 1460> (DF) (ttl 64, id 24607)
07:49:00.796367 (Houston Clt IP).3974 > (Boston SVR IP).23: . ack 1 win 8760 (DF) (ttl 128, id 48217)
07:49:00.803021 (Boston SVR IP).23 > (Houston Clt IP).3974: F 1:1(0) ack 1 win 33580 (DF) (ttl 64, id 24613)
07:49:00.803208 (Houston Clt IP).3974 > (Boston SVR IP).23: . ack 2 win 8760 (DF) (ttl 128, id 48473)
07:49:09.226096 (Houston Clt IP).3974 > (Boston SVR IP).23: F 1:1(0) ack 2 win 8760 (DF) (ttl 128, id 48729)
07:49:09.226151 (Boston SVR IP).23 > (Houston Clt IP).3974: . ack 2 win 33580 (DF) (ttl 64, id 29907)
```

Inside Interface:

```
Boston FW1 {9} % tcpdump -vni eb0 host (Houston Clt IP) and port 23
tcpdump: listening on eb0
07:49:00.800227 (Houston Clt IP).59985 > (Boston SVR IP).23: S 4021600304:4021600304(0) win 32768 <mss
1460,nop,wscale 0,nop,nop,timestamp 794222 0> (DF) (ttl 64, id 24609)
07:49:00.800784 (Boston SVR IP).23 > (Houston Clt IP).59985: R 0:0(0) ack 4021600305 win 0 (ttl 127, id 10659)
```

The **–n** switch tells TCPdump not to resolve IP addresses to names. The **–i** switch says to monitor traffic on the following interface. In the first terminal screen we will be watching the traffic on interface eb0, which in the case of the Boston Firewall is the internal interface, and the second terminal is eb1, which is the external interface. We ask the user to initiate a session with his application. In this example (Figure 1) we will use the telnet application. We watch both interfaces and see traffic on both interfaces and in both directions. This tells us that the Firewall is passing the traffic in this case and that everything looks good up to this point.

Figure 2 TCP Three-way Handshake

Client----------SYN----------→Server
Client ←----------SYN/ACK----------Server
Client----------ACK----------→Server

Figure 3 TCP connection close

Client----------FIN----------→Server
Client←----------ACK----------Server
Client←----------FIN----------Server
Client----------ACK----------→Server
(Either the Client or the Server can initiate a session close.)

Further analysis of the traffic taking place on the outside interface would indicate a complete session establishment and a clean session close. This is evident by a TCP three-way handshake (Figure 2) **appearing** to take place between the Houston client and the Boston Server. This is followed by a normal session close (Figure 3) initiated by the Boston Server. The reason the word **appear** was used is that if we take a look at the communication which takes place between the Firewall and the Boston Server we get a different picture. The original packet (with SYN flag set) from the client was passed on to the Server but the server returned a packet with a Reset flag. This is essentially a connection refusal. No other communication is required after a Reset. So what can we determine by all of this? Since this is a proxy Firewall, and an approved communication in the Firewall rule-set, the Firewall established a session with the client. It then attempted to make the connection with the server, which was denied. The Firewall then closed its session with the client. By monitoring the traffic on both interfaces we have discovered that the problem lies with the Boston Server itself. We would then concentrate our efforts on the Server configuration. Since the Server returned a reset flag we also know that this is not a user authentication issue. A session was never established in order to pass User authentication information.

Firewall Logs

One often-neglected feature available on most Application firewalls is their logging ability. Where else but at your domain gateway would be a more valuable place to keep track of who is entering and leaving your domain. What if your company gets hacked, everyone starts jumping around and pointing fingers at different areas as they try to piece together what happened. You receive a call from above wanting the answer several immediate questions, what happened? How did they get in. Oh, we forgot about the first question, how do we shut them down? Now, all of a sudden those neglected logs become extremely important. Not only might they help you find out how they got in, but they may help you shut them down. As said earlier, most Firewalls have some sort of logging ability and many of them have numerous configuration options to determine how much or how little logging to perform. Well, the more verbose the logs the better but there is a price to pay for this as well. You will take a hit in performance and drive capacity as well as an increase in administration due to the archiving required. It will be necessary to strike a balance. Just like backups, they aren't very important until something disastrous happens.

Conclusion

All of these things together make for a layered approach to Network Security. I think I might just look for another car to break into instead of the Explorer with the flashing red light on the dash the locked door, unknown cargo, and that big red Club attached to the steering wheel. I may be able to defeat all of those but why hassle with it when the one next to it has the keys in the ignition and that beautiful Kenwood CD player installed.

Nowadays all that I hear about Firewalls is their vulnerabilities. Or, that they are not the solution to all of our problems, that they even create problems. It seems that we are now swinging the pendulum the other way and trying to convince each other that a Firewall is not the solution. That, just because we have a vault at the bank, it doesn't mean we are safe. That may be true, but I plan on keeping my money in the bank with a safe, not the one without.

In the network where I work, we are a new breed. We were created with the sole purpose of looking at our network security. We do not have to balance between handling all of our network and connectivity issues and security. We are the full time Firewall administrators and network security analysts. Instead of taking our LAN Administrator and giving him a firewall and the additional duty of maintaining it we have created a new position: The Network security administrator.

The purpose of this paper is to hopefully bring the pendulum back to center. To balance between the catch-all Firewall solution and the waste of time Firewall plan. The Firewall is an invaluable asset to network security. Its success is limited only in how it is implemented. The time we spend on measuring and configuring the Firewalls, to balance

its strengths and weaknesses as the network changes with the times will determine how effective Our Firewalls and overall Network Security will be.

Bibliography

1. Hauben, Michael. Behind the Net – The untold history of the ARPANET. "History of ARPANET"
   URL: http://www.dei.isep.ipp.pt/docs/arpa--1.html (28 April 2001)

2. Avolio, Frederic. "Firewalls and Internet Security, the Second Hundred (Internet) Years".
   URL: http://www.cisco.com/warp/public/759/ipj_2-2/ipj_2-2_fis1.html (28 April 2002)

3. Ranum, M. and Avolio, F., "A Toolkit and Methods for Internet Firewalls," Proceeding of the summer USENIX conference, 1994

4. Wobus, John. "DHCP FAQ"
   URL: http://www.dhcp_handbook.com/dhcp_faq.html (26 October 1998)

5. Droms, R. "Dynamic Host Configuration Protocol". Request for Comments(RFC-2131), March 1997.
   URL: http://www.ietf.org/rfc/rfc2131.txt?number=2131

6. Egevang, P. Francis. The IP Network Address Translator (NAT). Request for Comments (RFC-1631), May 1994.
   URL: http://www.ietf.org/rfc/rfc1631.txt?number=1631

7. Srisuresh, P., and Egevang, K., "Traditional IP Network Address Translator (Traditional NAT)". Request for Comments (RFC-3022), January 2001.
   URL: http://www.ietf.org/rfc/rfc3022.txt?number=3022

8. Holdrege, M. and Srisuresh, P., "Protocol Complications with the IP Network Address Translator". Request for Comments (RFC-3027), January 2001.
   URL: http://www.ietf.org/rfc/rfc3027.txt?number=3027

9. "Why Can't Ipsec and NAT Just Get Along?" By Mike Fratto
   URL: http://www.networkcomputing.com/1123/1123ws22.html

10. Srisuresh, P., "Security Model with Tunnel-mode IPSEC for NAT Domains". Request for Comments (RFC-2709), October 1999.
    URL: http://www.ietf.org/rfc/rfc2709.txt?number=2709

11. "Teaching the Rules of the Road" By Jon Lasser
URL: http://online.securityfocus.com/columnists/77 (April 24, 2002)

12. Northcutt, Stephen and Novak, Judy. Network Intrusion Detection An Analyst's Handbook, Second Edition. New Riders Publishing, September 2000