Firewall And IDS Shortcomings

Char Sample, Mike Nickle and Ian Poynter

First presented at SANS Network Security, Monterey, California, October 2000

1. Introduction

Firewall and intrusion detection system (IDS) products continue to mature and to offer varying levels of functionality for the different aspects of Internet security. Current industry best practices recommend the use of firewalls and IDS products in concert with each other in order to have the most complete security picture. The concept behind this is to use the firewall for prevention and the IDS for detection, should the firewall be breached. In theory this sounds like a great idea; however in reality the idea is flawed. There are several reasons why reality does not match up with theory. The purpose of this paper is to explore not only this mismatch, but also specific product categories and to present the strengths and weaknesses associated with the various products.

Our conclusions are drawn from the evaluations that we have performed over the past few years, along with integration and support experiences. Each author has credentials in more than one area. Sample brings mostly firewall knowledge along with some IDS knowledge. Nickle brings mostly IDS knowledge and some firewall knowledge. Poynter brings the integration and support knowledge along with extensive experience with both firewalls and IDS products.

2. Firewall Shortcomings

The firewall shortcomings we have noticed are certainly not surprising. The very existence of the IDS products is in part based on the assumption that at some point the firewall will fail. While the time interval for this failure is undefined, the acknowledgment of such a problem would lead the casual observer to ask why. Some investigation reveals the causes, which include:

- Faulty design premise
- Faulty designs
- Platform dependencies
- The emergence of new applications
- Environmental problems

We will examine these causes in more detail below.

At the heart of most system problems is a faulty design premise, which typically manifests itself in the system requirements. Defining system requirements is not an easy task and attempting to do so in “Internet time” is downright daunting.

Firewalls are an interesting study in this problem. A long time ago when the Internet was young and not widely used commercially, the need for firewalls was not quite as urgent as it has become. The requirements were much looser: simply block the few troublesome networks (or addresses) and carry on. As things grew, this solution was widely recognized as inadequate; however the basic premise of blocking that which was not desired has remained. This premise led to the ”find and fix“ mentality that currently passes for best practices in the Internet security industry today.

The design premise of blocking undesired traffic would have been greatly improved had the designers thought to block unnecessary (anomalous) traffic instead. Some firewalls do this to a great degree (all the way to the operating system), others do not. Generally speaking, proxy firewalls do this more effectively than inspection firewalls but each firewall has its own shortcomings.

If we break the measurement criteria down to applications, rules and platform, we can more accurately quantify the problems of the firewall products. Let us take a moment to define the areas of interest.

- Applications deal with the commands being invoked by the application itself. These commands are specific to a particular application, such as: the GET, PUT and OPEN commands in FTP.
- Rules deal with the firewall rulesets. A sample rule might be “DENY from ANY to ANY on ANY port.”
- Platform deals with the operating system on which the firewall runs. If the platform itself can be toppled then there is a chance that the firewall can as well, since it resides on and is supported by the operating system.

2.1. Examining Firewall Products

We will examine each of these major firewalls in alphabetic order:

- Axent Raptor
- Checkpoint Firewall-1
- Cisco PIX
- Network Associates Gauntlet

2.1.1 Axent Raptor

Axent Raptor is a proxy-based firewall with packet filtering capabilities. The dedicated application proxies use a subset of commands and perform bounds checking, thereby making it very difficult to break the application. Threshold values are not typically checked making potential denial-of-service attacks possible. As new attacks are discovered there is a slim to moderate chance that they may actually work on some of the proxies, but if the attack requires

commands at the application level that are not supported by the firewall, then those attacks will definitely fail.

At the ruleset level things get a bit more complex. Depending upon the rules and how they are being used, a site may be vulnerable to data driven attacks in the best case, and in other cases they may be vulnerable to any number of attacks resulting from allowing access. The firewall has a generic proxy (called the GSP) that allows both TCP- and UDP-based traffic to be passed through without inspection. Newer versions also include the capability to allow purely IP-based traffic, opening up an array of possible problems for the inexperienced administrator. The GUI used for the Raptor is fairly intuitive and has improved in newer versions.

At the platform level, the Raptor firewall operates under the ”find and fix“ mentality. Operating systems are hardened to the extent that they remove unnecessary services and patch known vulnerabilities, but this firewall may sometimes be vulnerable to yet to be discovered vulnerabilities on the operating system. This is because the operating system is not secured from the ground up and the operating systems that Axent uses are not ”open.” It should be noted however that the Raptor does intercept network data at the IP stack, so network-based operating system attacks must come from trusted hosts.

2.1.2 Checkpoint Firewall-1

Checkpoint Firewall-1 is a stateful multi-level inspection based firewall with proxy-like capabilities built into it. True proxies are used when user-level authentication is required. The stateful inspection engine performs data inspection only on certain well-known protocols and allows good and unknown data to pass, while blocking known bad data. The data inspection does not use a subset of commands, but rather it uses the full command set, removing commands that are known to be bad. Thresholds are not checked, allowing potential denial-of-service vulnerabilities. New attacks that are discovered, particularly data-driven attacks, stand a moderate to high chance of being successful.

At the ruleset level, this firewall can be very intuitive with its easy to use GUI. Its main problem is the so-called hidden rulesets. Inexperienced users are often unaware of these built-in rules, which can cause unexpected behavior. These rules include permission for remote administration, along with rules to pass IP packet fragments. One of the default rules is a “deny all” rule, so that unauthorized traffic remains blocked. In order to use Firewall-1 in complex situations, a complete understanding of the inspection engine is required. [1]

---

[1] See Understanding the FW-1 State Table by Lance Spitzner at *http://www.enteract.com/~lspitz/fwtable.html* for an interesting analysis.

Firewall-1 does not harden the operating system at all when it is installed, so the installer must not only remove unused services, but also fully patch the operating system before installation.

2.1.3 Cisco PIX

The Cisco PIX is also a stateful multi-level inspection firewall with proxy-like capabilities built in. The implementation of data examination on the PIX is similar to the Firewall-1 (see above) and has the same issues.

At the ruleset level, this firewall has a “deny all” rule that keeps unauthorized users out. Data passed by authorized users is not always checked. Configuring the PIX without Cisco’s recently-released GUI is not for novice users and could lead to configuration errors.

Since the PIX is built on a version of Cisco’s IOS, the operating system is not hardened by the installer, but by Cisco. The vendor must release patches for the firewall and its integrated, underlying operating system.

2.1.4 Network Associates Gauntlet

The Network Associates (NAI) Gauntlet is a proxy-based firewall with packet filtering capabilities. The dedicated application proxies use a subset of commands and perform bounds checking, thereby making it very difficult to break at the application level. Threshold values are not typically checked, making denial-of-service attacks possible. As new attacks are discovered, there is a slim to moderate chance that they may actually work on some of the proxies, but if the attack requires commands at the application level that are not supported by the firewall then those attacks definitely will fail.

At the ruleset level, this firewall has never won high marks for ease of use. Even with the newer GUI, the conceptual framework must be clearly understood for correct configuration. Generic proxies (available for TCP only) render the protected systems vulnerable to data driven attacks, in the best case, and in other cases accessible hosts may be vulnerable to any number of attacks resulting from allowing access.

At the platform level, Gauntlet operates under the ”find and fix” mentality. The underlying operating systems are hardened to the extent that unnecessary services are removed and known vulnerabilities are patched. While the Gauntlet does perform a great deal of hardening on installation (which is especially true of its Unix versions), it can be vulnerable to as yet undiscovered attacks on the operating system. For many years the Unix version of this firewall ran BIND 4.9.4, even after vulnerabilities in this version of BIND had been discovered. In some cases the Gauntlet can withstand novel attacks on some services, but this has become less true over time. The operating system is not secured from the ground up, so the basic platform remains vulnerable to new attacks as they arise.

The packet filtering capability provides only limited logging capability and performs only limited checks. Its use is not recommended if it can at all be avoided.

3. Intrusion Detection System Shortcomings

We had hoped that the IDS products would bridge the gap resulting from the shortcomings of the various firewalls. Unfortunately this is not entirely the case. IDS products can be broken down into network-based (HID) and host-based (NID) products.

3.1. Network-based IDS

Let's talk first about network-based intrusion detection (NID) products. NID products work by listening to the network and examining the packets as they pass. Pattern matching is used to determine whether a packet matches all or part of an attack signature. This works very effectively against known attacks, such as an *rpc.ttdb* buffer overflow against Solaris or a *BackOrifice* connection attempt.

Most NIDs will allow a TCP connection reset to be issued against the offending connection, closing it down. If the offender is on a different network, many NIDs have the ability to add a rule to the access control lists on a router to prevent additional attacks from that host or network. For the most part, a NID system is a ”drop-in“ implementation that requires minimal effort from overextended system administrators and can easily be monitored remotely by security staff. This of course assumes that the volume of data produced by the NID does not overwhelm the staff monitoring it.

In spite of the great abilities of NID products they suffer from a basic design flaw: new attacks can slip through undetected, since they do not match any of the patterns currently defined. Some products allow for the creation of new signatures, but this not only requires a high level of expertise in defining them, but also a level of abstraction on behalf of the operator. Overworked operators and administrators either do not have the skills to add new attack signatures by themselves, or simply do not have the time to do this.

Three obstacles that plague NID products are:

- Line speed. The products simply cannot keep up with the volume of data on the network.
- Switched environments. NIDs must be carefully placed in a switched environment, and place load on the switches to deliver data to the spanning port.
- Encryption. No NID can review encrypted data, since the keys are not available. This allows attacks hidden in encrypted connections to proceed without detection.

As we examine each product, we will see that these obstacles will not only continue to plague this line of product, but in some cases (like encrypted connections) will never be solved. The other obstacles will either give rise to a new generation of products, or current products will be redesigned in order to overcome them. A recent MIT study funded by DARPA of the various NID products found that they typically only detect between 60 and 80% of attacks that take place across the wire.[2]

3.1.1 Line Speed

The line speed problem probably receives the most press and rightly so, since it is one of the major selling points that NID vendors use to differentiate each product from a crowded marketplace. Several vendors claim that their NID can perform traffic analysis in real-time at wire speed. This is simply not true for those systems that operate on top of an off-the-shelf operating system (Solaris, NT) due to the resource constraints that are imposed by the underlying operating system. Systems built on a general-purpose operating system are rather flexible in how they can be deployed and are easy to update for new attack signatures or improved scanning algorithms, so they are still attractive where the bandwidth of the network to be monitored is limited.

Embedded systems, such as those that are programmed directly into an ASIC for example, operate at extremely high speeds but are viewed as difficult to update and rather rigid. We expect that these systems will improve over time and that more hybrid systems combining the benefits of both embedded and general-purpose operating system approaches will emerge.

3.1.2 Switched Environments

The switched environment problem results from many of the same forces behind the line speed problem. A NID must be able to "hear" a known attack in order to be able to analyze it. Therefore the NID would have to listen to every segment connected into the switch. This can require massive computing resources when a general-purpose operating system-based NID is used on a typical enterprise switch.

The obvious solution would be to embed NID technology directly into the switching equipment itself at the backplane level. This evolution in devices has been anticipated for quite some time (at least since Cisco has acquired WheelGroup) but it is only just beginning to become available. Nevertheless Cisco and other vendors have a challenge on their hands in trying to embed the NID while keeping it flexible and maintainable.

---

[2] Evaluating Intrusion Detection Systems: The 1998 and 1999 DARPA Off-line Intrusion Detection Evaluations. Richard P. Lippmann et al. DISCEX Presentation and Paper January 2000.

3.1.3 Encrypted Connections

The encrypted connection problem exists due to the inability of the NID to decrypt the sessions that it is recording, much less to decrypt them in anything near real-time, even if keys were available. The bulk of encrypted traffic on networks currently is either SSL sessions, VPN products linking networks or mobile users, or users connecting via SSH.

In order for a NID to analyze an encrypted session, the product would have to:

- Be programmed with keys to a session or somehow "grab" keys that are transmitted while the connection is being built (which we hope is not possible in a well-designed encryption algorithm)
- Know the encryption algorithm being used for session, to allow the proper decryption to take place
- Understand the application protocol. This is at least a less insurmountable obstacle than the previous ones
- Capture and decrypt the entire session in real time
- Guarantee to network managers that there is no way to compromise the NID host

Because of these barriers, the encrypted connection problem is currently insurmountable by any NID. This has led to the rise in host-based intrusion detection (HID) products. The basic approach of the HID products is surprisingly very proactive (surprising because the security industry as a whole is very reactive and this is reflected in the industry products).

3.1.4 ISS RealSecure

ISS RealSecure from Internet Security Systems is a NID product that integrates directly with ISS's HID offering. While this integration between the products has its advantages, RealSecure has some major limitations. Among the most obvious limitations are its high false alarm rate and its limited support for customized alarms. We know of very few enterprise environments that are running completely standard services and do not have at least some use for custom signatures.

RealSecure benefits from having an extremely straightforward GUI and its ability to integrate into the Tivoli and HP OpenView network management systems. NT-centric shops will appreciate that the sensor is available for Solaris and NT and that the HID product is also available for NT.

3.1.5 Cisco NetRanger

Cisco NetRanger is one of the most mature products in this space. This product was originally developed by the WheelGroup, which was acquired by Cisco in 1998. The NetRanger sensor is a PC running Solaris x86 with minor tweaks to

optimize it for its task. Installation of the sensor takes about 15 minutes once the system is taken out of the box and connected via the included console cable. The sensor is extremely fast and has proven to be quite reliable. Unfortunately, Cisco enables telnet and FTP access to the sensor rather than implementing secure access.

Where NetRanger falls down is in the management console and HID integration. NetRanger does not have its own management console; instead the user must purchase a copy of OpenView (with a version number below 6.10) and use that as the primary interface. Cisco does not currently offer integration support with any HID product. Hopefully Cisco will narrow this gap and allow the product to move forward. On the plus side, Cisco has done a good job in its support for changing access control lists on Cisco routers dynamically and directly accepting logs from the PIX product. It should be noted that vendors have more incentive to interoperate with their own products, so we don't expect a high degree of integration for third party HIDs or firewalls.

3.1.6 Network Flight Recorder Intrusion Detection Appliance

The NFR Intrusion Detection Appliance (IDA) from Network Flight Recorder combines the simplicity of a fully "appliantized" sensor and web-based management console with the power of a flexible scripting language. The scripting language allows the definition of attack signatures that are truly tailored for a particular network. Of course the expertise must be available to define these signatures, which is often a sticking point.

The price point for NFR is extremely aggressive, which means that it can be more widely deployed than some of its competitors, especially in cost-conscious organizations. The sensor system is booted from a CD-ROM, which prevents molestation by intruders and eases upgrades. An upgrade for NFR merely involves replacing the CD-ROM and rebooting.

While the NFR product performs well, it still suffers from the same fundamental flaws as other NIDs: encrypted sessions, switched environments and attack signature accuracy. Some users will find the volume of variables that can be adjusted to be somewhat dizzying, although it must be said that this is true for all of the IDS products.

3.1.7 Axent NetProwler

Axent NetProwler is a NID that is primarily intended to support their excellent HID product, Intruder Alert. This product is not quite as mature as some of its competitors but benefits from good engine speed and good logging functionality. Unfortunately, its detection capability needs further refinement and it is not quite ready for enterprise-wide deployments. For an Axent customer who has already made a heavy investment in HID, it is definitely worth considering.

3.2. Host-based IDS

The HID products with their proactive approach would seem to be the missing piece to the firewall-NID puzzle; however at the system level, there is little to no reconciliation of the network and host findings. Also there are several major obstacles to HID systems. These include

- Real-time response
- System resources
- Correlation and reconciliation

3.2.1 Real-time Response and System Resources

Whenever the HID springs into action, it takes away resources from the rest of the system. Therefore system operators run many of these products in a batch mode in order to minimize the impact on operations. This obviously creates a huge issue for the concept of real-time processing. Then the resource issue comes into play. Constantly scanning a file system or piggybacking every network connection takes up a considerable amount of resources. For example, with a network connection, it approximately doubles the resources required. For you're the average eCommerce server, this additional resource overhead is just too much.

3.2.2 Correlation and Reconciliation

Then come the correlation and reconciliation issues. How can the HID data be correlated with data from the NID systems? How are reports and alarms reconciled against normal system use?

In most cases this must be performed by hand, with the notable exception of ISS. The ISS products in NID and HID areas integrate with their decision support tools, although there are some security layering issues that arise from this. Several interesting white papers have been written about automating this process, but no one has tackled integrating firewalls, NIDs, single sign-on systems, HIDs and all of the miscellaneous security components into a comprehensive management and reporting system. We suspect that this is too complex a problem to be completely or even adequately solved any time soon, although we're sure that vendors are trying.

3.3. Common Vulnerability and Exposure Project

Mitre Corporation recently announced (in October 1999) the Common Vulnerability and Exposure project (CVE) in order to improve event correlation across systems. Several of the major security vendors, including Axent, Cisco, ISS, Symantec and NFR are participating. This initiative will hopefully provide us with a common language for labeling exploits. Unfortunately very few

vendors are shipping products that provide CVE output at this juncture. Furthermore there seems to be a lack of products in the industry that collect all of the CVE output and correlate it in real-time (or near real-time).

4. Example Scenarios

All of the obstacles discussed above lead to the inescapable conclusion that we are spending quite a bit of money to arm ourselves, but yet we are still not fully protected. In fact we are, in many cases, vulnerable to the next attack that emerges even if we diligently apply all the currently available security tools. Below we walk through some scenarios that illustrate how different attacks could succeed in the ”best practices“ security environment. This environment includes a perimeter firewall, an intranet firewall, some NID products on ”key networks“ and HID products on ”key servers.”

4.1. Scenario One

This scenario details an exploit through a partner network, using a host at the site as a relay in an attack.

In figure 1 we see an attack scenario that is becoming somewhat common. An attacker will leverage trust relationships between networks to defeat network defenses that are functioning normally. In this case the attacker first penetrates the application service provider shown (Mega ASP). Once inside the ASP’s network, the attacker discovers that several of the servers have some of their routes set to a VPN gateway, rather than the default route through the firewall. Based on this, he begins the process of slowly probing the customer network (*burneduser.com*). Even though the customer has been diligent in securing their network, they’ve turned down the alarm thresholds for traffic originating from the ASP, since they are a “trusted” network and they perform remote maintenance on equipment on the customer’s premises.

Once the attacker finds a few interesting systems, he leverages the information gathered from the ASP servers to mount his attack. On the second try, he grabs the password database from one of the servers and begins decrypting it using *John the Ripper*. In a few hours, he’s impersonating users and viewing proprietary information and sending salary figures to all employees and a couple of competitors.

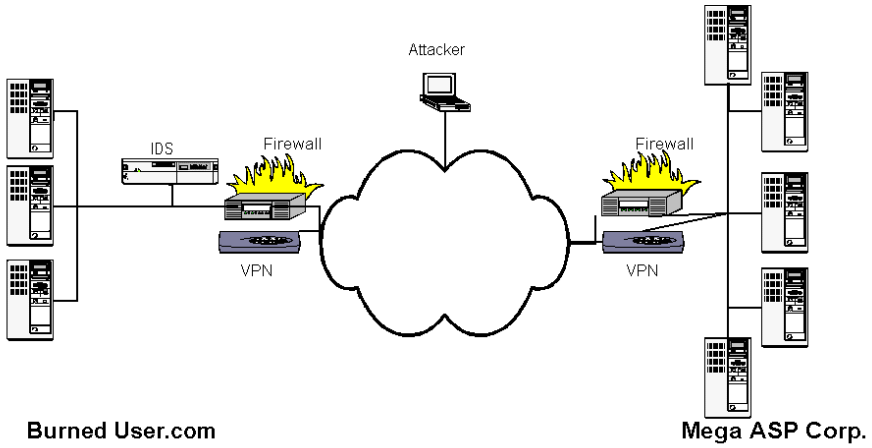

Figure 1.

4.2. Scenario Two

This scenario details an exploit through an eCommerce application in order to gain internal access. In this case, the attacker jumps to another internal host that has legitimate access to a trusted server. In this scenario, detailed in Figure 2, the victim has a relatively sophisticated eCommerce implementation with layered perimeter security and intrusion detection.

The environment consists of IIS front edge servers that speak to a back-end using SQLnet connections that are brokered via Tuxedo. There is a firewall in front of the edge servers (PIX) with a RealSecure IDS system listening to the uplink that connects to the firewall in front of the database (Checkpoint). There is Tuxedo TP monitor server, which was hardened by the developers, straddling both the internal (trusted net) and the edge net.

The intruder breaks into the IIS boxes with an HTTP buffer overflow and uses "netstat –a" to find many connections to database servers through the Checkpoint and one or two connections to the Tuxedo box.

After taking a stealthy look at the firewall which serves that internal route, she uses *nmap* to scan the Tuxedo server and finds an out-of-the-box installation of Solaris with telnet and ftp turned off. Apparently the developers were unaware that a dual homed Solaris box will run *routed* making it an easy task to gather routing information and to connect through the machine. Since there are two network interfaces, each on a separate network, *routed* is running by default. This mistake would have been caught if operations were able to check policy against this machine but the eCommerce group is in control.

Our intruder changes her route from the firewall to the Tuxedo box and bypasses the IDS, subsequently launching a successful attack against the internal hosts.

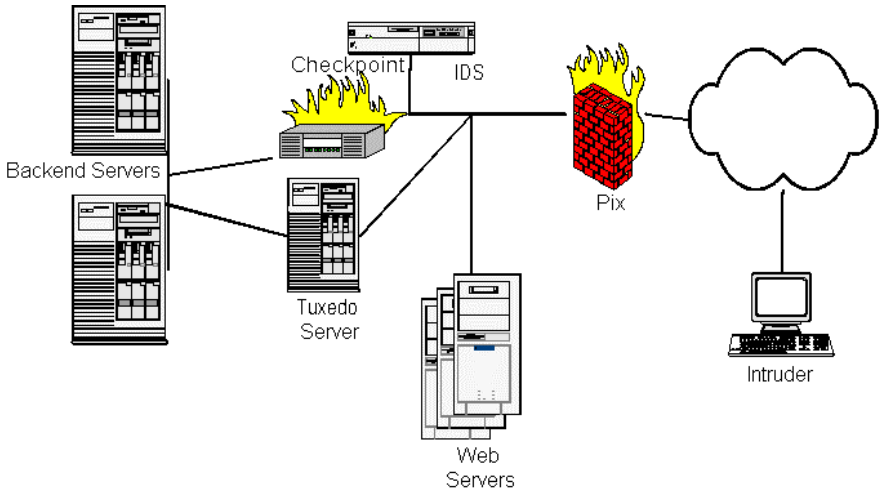

Figure 2.

5. Conclusions

Our example scenarios bring into stark relief the unintended consequences that can come from a complex implementation. The weakest link in any architecture will be the one that is ultimately exploited by an attacker.

While firewalls and both types of IDS do offer some protection, they do not offer the level of protection that would make many security ”purists“ very comfortable. Still, this does not mean that we should throw out existing equipment, but rather that we need to demand better and more robust product features, and we should demand that the security vendors' products be enhanced to reflect a more comprehensive design.

Furthermore we need to ensure that we communicate with all of the stakeholders on any project involving the network. Only through improved products and effective communication will we be able to create effective systems for stopping “the wily hacker.”

5.1. Sources

Intrusion Detection, Take Two, Network Computing Magazine, November 15, 1999.

Network Intrusion Detection, Stephen Northcutt, New Riders Press 1999

Understanding the FW-1 State Table, Lance Spitzner, *http://www.enteract.com/~lspitz/fwtable.html*.

Evaluating Intrusion Detection Systems: The 1998 and 1999 DARPA Off-line Intrusion Detection Evaluations, Richard P. Lippmann et al., DISCEX Presentation and Paper January 2000.