A Review Of Floppy-Based Firewalls And Their Security Considerations

Sean Closson
GSEC Practical v1.3

Abstract

This paper is for the user that is evaluating inexpensive perimeter firewall solutions. Several distributions of miniature Linux systems are available for repurposing old computers into valuable firewalls and routers. There are many advantages in selecting one of these distributions for your firewall project, and this paper discusses the features and security implications amongst three of the more popular choices available.

After reading this paper, the user will have a better understanding of floppy disk-based firewalls and some of the technologies they employ.

Introduction

Security is a pressing issue that all network administrators, indeed *all* users must face. There are numerous tools and technologies that can be employed to increase a network's security posture. One of the most common technologies is a perimeter firewall. There are a multitude of commercial vendors that offer firewalls and some that market their devices as firewalls. Of course firewalling technology is available in GNU/Linux and BSD distributions as well. With so many choices, how can someone with a limited budget or limited technical ability utilize a firewall? Enter the floppy disk-based Linux firewall.

The Linux community has a history of doing amazing things with limited resources. The floppy-based firewall is the epitome of this trait. Developers have managed to fit a fully functional operating system, network interface card (NIC) drivers, packet filtering rules, and more onto a single floppy disk. Adding to their value, many of these distributions provide an easy to use script that will prompt you for your desired configuration and build the floppy image for you. By doing this, the developers bring the power and flexibility of firewall technology to people lacking the technical skills or information to build a firewall from scratch. Before you spend the money on a hardware firewall, take the time to evaluate the capable firewalls discussed here.

This paper isn't a "How-To" on making or installing floppy disk-based firewalls. It does not contain feature comparisons with commercial hardware (such as Nokia, Cisco PIX, etc.) or software (such as CheckPoint FW-1, NetGuard Guardian, etc.) firewalls.

Why floppy-based Linux?

There are many reasons to use a floppy-based firewall:

- Utilizes hardware that may otherwise be obsolete
- Free to download
- Relatively easy to create and use
- Small footprint, often no unnecessary services & utilities
- Expandable, offering additional services
- Solid state, no slow disk access once loaded into the RAM drive
- Offers basic perimeter security
- Community support, most often in forums
- Opportunity to learn more about firewall technology

One aspect of floppy-based firewalls that is not often talked about is that by simply swapping out the floppy *you can completely change your firewall*. Imagine the overworked network administrator that is charged with implementing a firewall solution, but is not allowed to experiment with the production device. Ideally, the administrator would build another box with the same network cards for testing and emergency use. In a lab environment the administrator can build the floppy, add filtering rules, and test it before putting it into production. When it is time to "go live", the administrator puts the new floppy in and reboot.

Of course there may be compelling reasons to avoid using a floppy-based router: prohibited by existing security policy, vendor affiliation, or the need for more advanced features such as intrusion detection.

One option available for people building their own firewall is to build it on a full operating system installation. Doing so adds complexity, the potential for numerous vulnerabilities, requires specialized skills (or at least patience and perseverance), and requires a hard drive. To address these issues, developers have created specialized feature sets and fit them onto a bootable floppy disk.

What security do these firewalls provide? In order to answer that, a basic understanding of some key concepts is required.

Concepts

This paper contains terms that, unless defined in context, may be subject to interpretation and challenge. For example, when a colleague, sales person, or author of a news article uses the term *firewall*, they may not all mean the same thing.

Network Address Translation (NAT)

Network Address Translation, also referred to as masquerading, is the technique for hiding one or more IP addresses behind another. The most common use for this is when following RFC 1918 [5]. The RFC was written to address the looming shortage of available IP addresses. Certain blocks of networks were designated as private, set aside for networks to use as their own as they saw fit. The RFC states that these addresses should not be routed in order to prevent ambiguous routes and confusion.

Port Address Translation (PAT)

Looking at layer 4 of the ISO model (where TCP/UDP is handled), PAT enables the firewall to forward packets received on one port to a different TCP or UDP port on another host. In fact, the firewall may forward packets received on one port to a different host than those received on another port. The net effect of this is that all services handled by several internal hosts actually appear as if they were all on the edge device.

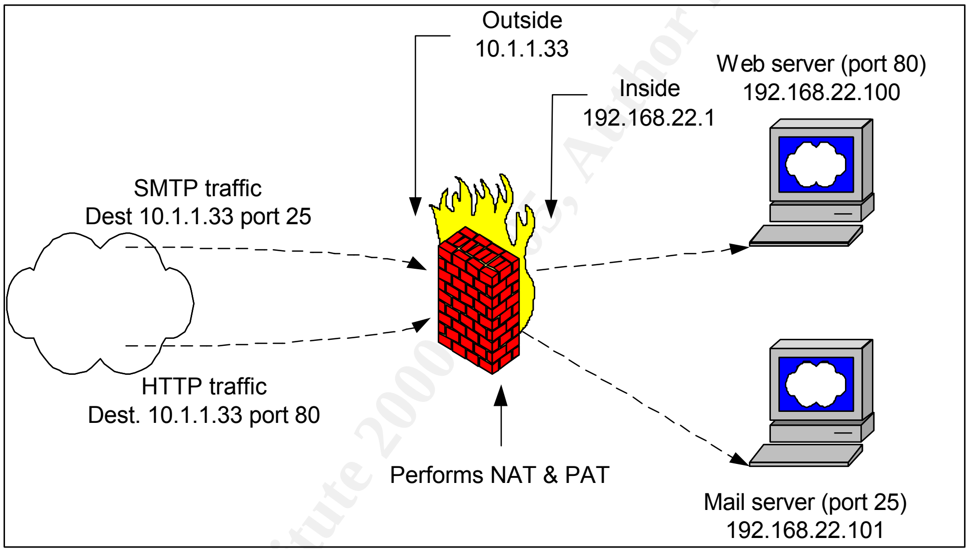

Figure 1 - Diagram of NAT & PAT Traffic Flow

Router

A router, in its simplest usage, gets packets from one interface to another. It does this by looking at the header of the IP packet and basing decisions on the destination address found there. Some routers can accept or deny packets based on access control lists (ACL's) and are often referred to as "screening routers." Of course many modern routers offer more features, but discussion of those features are outside of this paper's scope.

Firewall

Of all the commonly used security terms used, this one is the most nebulous. In the book *Building Internet Firewalls* (O'Reilly), the authors equate a firewall to the moat of a medieval castle [1]. The two, they explain, serve multiple purposes:

- It restricts people to entering at a carefully controlled point.
- It prevents attackers from getting close to your other defenses.
- It restricts people to leaving at a carefully controlled point.

Ask a security professional what a firewall is and you may hear about NAT, PAT, and packet filtering. Another security professional may decry that a device can not be considered a firewall unless it includes extensive logging and stateful inspection. For the purpose of this paper, the term firewall will mean a device that performs packet filtering, logging, and either a supports a third interface for the DMZ or employs port forwarding (PAT).

Packet filtering vs. stateful inspection

What is packet filtering exactly? How does that compare to stateful inspection? Packet filtering at its most basic level allows or blocks packets based on source address, destination address, and the service port. By only looking at this part of the packet headers, packet filters or screening routers, can provide basic security with high performance. Stateful inspection takes this one step further and allows the firewall to only allow inbound packets that are in response to an outbound packet. To do this, the device must maintain the "state" of every packet, which adds to the device load and memory requirements.

Proxies

A proxy-based, or application layer gateway, firewall acts as an intermediary for the network traffic. To both computers trying to communicate, it appears that all traffic is to/from the firewall only. Since the proxy works in the application layer of the OSI model, it is inherently more processor intensive than other methods. A proxy is programmed for a particular application layer protocol. Popular protocols like HTTP, FTP, and SMTP are easy to find proxies for. Custom programs may not have proxies built and therefore you may not be able to proxy that protocol. However, chances are good that someone else has dealt with a similar program already and has built a proxy for it.

Common distributions

While not a comprehensive list of available floppy-based firewall distributions, the following are easily found and under active development. Most run on a 386 class computer with as little as 8 megabytes of RAM. Since performance is partially dependent on the hardware platform, the more powerful the processor the faster it can process filter rules. Since all these distributions are loaded from a floppy disk into a RAM drive, you will be able to support more functions if you have at least 16 megabytes of RAM. Obviously, the firewalls require at least two interfaces to pass traffic. For an Internet connected firewall, an older 10 Megabit network card is more than sufficient. All three of the researched distributions use network address translation to provide a basis for security. PAT must be manually enabled if you elect to provide publicly available services.

Testing methods

In order to provide worthy conclusions, a consistent testing method is required. This should not be misconstrued as a scientific test. Although advanced firewall testing was beyond the scope of this document, an article written by Dr. Eugene Schultz [15] was invaluable in understanding aspects of formalized firewall testing. For testing purposes, a lab environment was created to provide isolation and repeatable scenarios.

Network topology

The test network consisted of three computers and two Ethernet hubs. Two computers remained the same throughout the tests, emulating the attacker (named "snoop") and the protected host (named "target"). The third computer was the firewall device.

The target computer hardware boasted an AMD Athlon 900MHz processor, 512MB of RAM, and an Intel Pro/100+ network card. It ran a default installation of RedHat Linux 7.2 with the notable addition of Nessus. During outside penetration tests, TCPDump was used to detect leakage through the firewall.

The firewall machine was selected to be representative of older equipment that would typically be used with a floppy-based firewall. The computer was an older HP Vectra Pentium Pro 180, with 24MB RAM, one Linksys PCI NIC, and one 3Com 3C595 PCI NIC.

The attacking machine was identical to the target computer.

Tools

The tools used were selected for their technical abilities and commonality. While not the only tools used by hackers, they have become the de facto standard for certain aspects of security testing.

NMAP

Written by a self-proclaimed hacker [12] who goes by the alias Fyodor, nmap is arguably the most common port scanner and often the first tool used in an attack. Nmap enables many advanced scanning techniques to be used in penetration testing of firewalls. Testing was done with both SYN scans and XMAS scans using the following commands:

```
Nmap –v –sS 192.168.111.1       <-SYN scan
nmap –v –sX 192.168.111.1       <-XMAS scan (FIN, URG, & PUSH flags set)
```

Nessus

Written by Renaud Deraison and Jordan Hrycaj [11], Nessus builds on nmap by providing specific vulnerability testing. Tests were run using the "enable dangerous tests" setting from both the inside and outside interfaces of the firewall. Hundreds of scripts are available for testing almost any application in common use.

FreeSCO (0.2.7)

According to Serge Storozhevykh, the creator and main developer, FreeSCO stands for Free ciSCO [13]. It is touted as an alternative to traditional hardware routing products from well-known vendors. Based on kernel version 2.0.38, it offers Ethernet/dialup/leased line routing, supports up to three Ethernet segments, and up to two analog modems. Additionally, it provides time, DHCP, print, telnet, and HTTP server daemons. Routing is static; no dynamic routing protocols are supported.

Assuming the hardware is assembled, creating the firewall can be done in only a few minutes. After downloading the image and creating the floppy disk, just boot your machine with the disk and the program steps you through all the configuration settings. Most questions are multiple choice. When you have answered all of the questions, it will save its configuration and reboot. At this point you have a basic firewall at your perimeter. Depending on your needs, additional configuration may be desirable.

One nice feature of this distribution is the web interface you can use to administer your firewall. Although there are some limitations, you can enable/disable any of the services, review statistics, and even reboot it remotely. It also offers the ability to submit single-line commands without using the full command line interface. There is a menu-based system available for administering the firewall from the console or via telnet, giving you full control and ease of use.

Of the three distributions reviewed for this paper, FreeSCO is based on the oldest technology. Using kernel 2.0.38, the firewall rulesets are based on ipfwadm, the precursor to IPChains. While not as robust as its successors, ipfwadm does protect against [2]:

- IP spoofing - this is where a host from the public side sends out packets which claim to come from an inside trusted host.
- Source Routing - where an intruder mimics an IP packet coming from a trusted system.
- It can explicitly deny specific hosts from accessing services as well as trusted hosts from accessing un-trusted sites.
- By directing flow of ip packets be it tcp, udp, or icmp. It can control the flow of data in any direction customized depending upon your specific needs.

The built in web server, which can be used for a publicly accessible or an internal administrative web server, is thttpd v2.05, which contains several known vulnerabilities [6]. If it is important to use FreeSCO *and* the web server, ACME Laboratories, the creators of thttpd, offer version 2.20c for download [3].
For help with FreeSCO, the forum available at http://www.freesco.net/cgi-bin/forum/UltraBoard.cgi is a valuable resource.

Coyote (1.2)

According to Coyote's FAQ [3], "Coyote was originally derived from LRP (the Linux Router Project), but has undergone over a year and a half of changes. Although Coyote still has some of LRP's base scripts, for the most part it is no longer compatible with LRP."

This distribution is based on kernel version 2.2.19 and utilizes IPChains for the firewall rules. Features include DHCP server & client, static routing, PPPoE support, PPTP pass through, and PPP dialup.

To create the floppy disk, you have a choice of two methods. For users of Windows systems, you can download and run the wizard, which provides an attractive GUI and prompts you for a few basic configuration settings. It will then create the floppy image for you and copy it to the disk. However, it is only capable of the most rudimentary of configuration; mainly network card settings. To modify firewall rules, you must edit text files on the floppy disk.

The other method is for users of Linux systems and offers a few advantages over the wizard method. By running the makefloppy.sh script, you can create custom images and include SSH for remote administration. This method provides the most flexibility in creating the firewall and its corresponding feature set.

Remote administration is done using telnet by default, although SSH is available if you build the floppy using the Linux method described above. Coyote is very extensible, utilizing the popular LRP package system. Packages are available for download if you desire to add web-based administration of the firewall, but are not included by default. Packages are available to add functionality to your Coyote firewall, from basic SSH to 3rd party software like Seattle Firewall [8].

NAT is enabled by default and PAT is possible by adding appropriate ipmasqadm rule sets using a menu system, or editing a file on the disk. VPN capabilities can be added using the IPFWD package and are limited to passing PPTP only. Using PPTP as a client or server on the firewall itself is not supported. This may be sufficient if you are using a Windows server running PPTP, or PoPToP [4] on a Linux box on your internal network and have added the appropriate ipmasqadm rules. Administrators should be aware of the security implications inherent with PPTP versus other VPN technologies [14].

The Coyote firewall provides good security and uses the established IPChains packet filter technology. Specific rules can be added to the default configuration that will enhance the firewall's security posture. While testing for vulnerabilities using Nessus, the only items of note were related to predictable IP sequencing numbers. While not a security "hole", an attacker could leverage that information for a man in the middle exploit.

FloppyFW (1.9.18)

Thomas Lundquist, author of FloppyFW, describes it as "… a static router with the firewall-capabilities in Linux." [9] The author created this distribution primarily for his own use, but has found that other people were benefiting from it. Some people have even written custom packages for it to address specific needs.

FloppyFW, version 1.9 offers the 2.4.17 kernel (and therefore IPTables) and adds stateful packet inspection. IPTables offers many improvements over earlier technologies such as IPChains and ipfwadm. For a quick feature comparison of IPChains versus IPTables, Josh Ballard's Oofle.com Firewall Center is highly recommended [7].

FloppyFW is not the easiest distribution to install. It requires the use of rawrite or WinImage (a GUI disk image manager) to create the disk. Configuration changes are done by editing text files on the floppy with another computer. The distribution does not have a built-in editor to handle this. As always in network security, usability and security are a tradeoff. While it may be inconvenient, this does offer some measure of protection. If the firewall were to become compromised, the attacker would find it difficult to make permanent changes with the sparse utilities available.

Remote administration of FloppyFW is not possible without the addition of third party packages. Developers have released numerous packages that will add a variety of functions to FloppyFW; some of these may offer secure remote administration.

FloppyFW boasts the most current kernel of the three floppy-based firewalls reviewed, and in turn enjoys the advantages of IPTables. IPTables replaced IPChains as the packet filtering technology used in Linux, and offers finer granularity for rules. Using IPTables, rules can now filter by MAC address, provide Type of Service prioritization, and all six TCP flags can be inspected instead of just the SYN flag as in IPChains [10]. Perhaps the most talked about new feature in IPTables is stateful inspection. This allows the firewall to make filtering decisions based on more than just what is in the current packet. It maintains the state of communications and therefore can allow packets through that are in response to a connection initiated internally. FloppyFW provides NAT, rather it provides both Source NAT (SNAT) *and* Destination NAT (DNAT). This feature is a product of IPTables and may offer you new techniques for address translation.

The internal scans against FloppyFW only revealed ICMP timestamp answering, and predictable IP sequencing. Examination of available utilities in the run-time file structure indicated a reasonably secure environment. It is obvious that efforts were made to provide only the minimum amount of functionality required to perform firewall duties. This philosophy is often used in secure operating systems and devices.

Conclusions

Overall, the power that each of these distributions provides is impressive. All of them employ some sort of package system to add new abilities that enhance existing functionality. Of course, selecting the best choice depends more on the needs of the end user than a list of features. Each of the reviewed distributions has characteristics that may benefit some users more than others. All provide *some* security, which is better than *no* security. Upon closer inspection, the differences between the distributions become apparent.

FreeSCO is the easiest to configure for installation and administer in production. However, it provides services that are not typically run on a firewall; like print and time services. The reliance of remote administration by clear text telnet and the vulnerable thttpd web server lessens FreeSCO's appeal as a security solution.

Coyote is extremely easy to build for Microsoft Windows users by utilizing the Windows wizard utility, but the wizard does not offer much configuration beyond that of network cards. The distribution does offer improved security over FreeSCO and has a more modern method of packet filtering. IPChains is a well-established technology with plenty of available support. The use of telnet for remote administration, albeit on the internal interface only, should be considered an inadequate method. Secure Shell is available as a third party package, and its use is strongly encouraged.

FloppyFW is the most difficult to create, configure and administer. However, with the most modern kernel and packet filtering technology it has the most potential as a security solution. IPTables offers new capabilities and improved packet filtering for firewalls, and will continue to be supported for the foreseeable future. Security is enhanced by FloppyFW's minimalist approach; there are no unnecessary services running, no remote administration channels to be compromised, just the bare essentials to get the job done.

For the quickest, easiest installation, try FreeSCO. With its prompt-based configuration, you can have a functional firewall in a little as five minutes. However, be aware that it may not provide you with much more than basic security.

For the best possible security provided by a floppy-based firewall, FloppyFW is the distribution to use. The minimalist installation and the use of IPTables offer the best security available on such a space-constrained medium. Be willing to put in some extra effort during installation and configuration in order to get this firewall distribution running properly.

References

1. Zwicky E, Cooper S, Chapman D. Building Internet Firewalls, Second Edition. Sebastopol, CA: O'Reilly & Associates, 2000. 21
2. dreamwvr. "IPFWADM FAQ - Frequently asked questions." URL: http://www.fwtk.org/ipfwadm/faq/ipfwadm-faq-2.html#ss2.1 (3 Mar. 2002)
3. Da LAN Tech. "Coyote Linux FAQ." URL: http://www.dalantech.com/coyote-faq.shtml (26 Feb. 2002)
4. Ramsay, Matt. "PoPToP - The PPTP Server for Linux." 11 Oct, 2000. URL: http://poptop.lineo.com/ (1 Mar. 2002)
5. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E. "Address Allocation for Private Internets." Request For Comments: 1918. Feb. 1996 URL: http://www.rfc.net/rfc1918.html
6. Gobbles. "thttpd Basic Authentication Buffer Overflow Vulnerability." 23 Nov. 2001. URL: http://online.securityfocus.com/bid/3562 (2 Mar. 2002)
7. Ballard, Josh. "IPTables vs. IPChains Comparison." 2001. URL: http://www.oofle.com/iptables/comparison.htm (5 Mar. 2002)
8. Eastep, Tom. "Installing Seattle Firewall on the Coyote LRP Distribution." 10 Jun. 2001. URL: http://seawall.sourceforge.net/Coyote.htm (27 Feb. 2002)
9. Lundquist, Thomas. "floppyfw." 1 Mar. 2002. URL: http://www.zelow.no/floppyfw/ (2 Mar. 2002)
10. Napier, Duncan. "IPTables/NetFilter — Linux's Next-Generation Stateful Packet Filter." URL: http://www.samag.com/documents/s=1769/sam0112a/0112a.htm (5

Mar. 2002)

11. Deraison, R., Hrycaj, J. “Contacts.” Nessus web site URL: http://www.nessus.org/contact.html (1 Mar. 2002)

12. Fyodor. “Who is Fyodor?” 26 Jul. 2001 URL: http://www.insecure.org/myworld.html (4 Mar. 2002)

13. Storozhevykh, Serge. “WWW.FREESCO.ORG” 17 Jun. 2001. URL: http://www.freesco.org/ (26 Feb. 2002)

14. Marcotte, Greg. “Protocols serve up VPN security.” Network World, 31 May 1999. URL: http://www.nwfusion.com/news/tech/0531tech.html (5 Mar. 2002)

15. Schultz, E. Eugene, Ph.D. “How to Perform Effective Firewall Testing.” 1996. URL: http://www.linuxsecurity.com/resource_files/firewalls/how2test.htm (5 Mar. 2002)