Global Information Assurance Certification Paper

Linux: A powerful force at the network perimeter.

Richard Johnson

GSEC Practical: Version 1.4b (Option 1)

Introduction.

The network perimeter is your first line of defense. It separates your network from the outside world. You need powerful security tools and a secure architecture to protect your network. You have countless options at the perimeter, depending on the type of network you want to protect, the defensive posture you choose, and the services you wish to offer to your users. With all the variables to consider, one thing is certain, there is a Linux based solution for nearly every function on your network perimeter.

With this paper I will discuss my belief that an investment in Linux will yield great returns in the future and security of your network. I will relate some of my experience as a Windows network administrator who has decided to explore new ground in the deployment of a Linux based perimeter. I will examine some of the hardware, software, and topology issues involved with perimeter design and I will illustrate a good way to get started for those like myself who are new to Linux.

A Windows admin (finally) takes a hard look at Linux.

Over the last few years I have been very aware of the emergence and progress of the Linux operating system. I subscribe to many of the trade magazines and there are always articles about the growth and advancement of Linux in the industry. If you browse the websites for these trade publications (www.eweek.com, www.infoworld.com, www.computerworld.com, etc.) you will see what I mean. The open source Operating System (OS) has become very well established in the server market (with support from IBM and Dell for example) and will someday be a real contender in the battle for the desktop OS supremacy, giving Microsoft a run for its money.

> Linus Torvalds, who created Linux in 1991, called desktop Linux "inevitable." "We already have all of the tools, in open-source software, necessary for 80 percent of office workers in the world: an office suite including spreadsheet, word processor, and presentation program; a Web browser, graphical desktop with file manager, and tools for communications, scheduling, and personal information management," Torvalds said in a statement. (Weiss, Todd)

I had read a lot about the various tools available but having no knowledge of Linux (or UNIX), I never really put much thought into investing time or money in the OS. That is, until I started being more responsible for the network and security in my organization. As a Windows guy, I looked at my perimeter and immediately thought about all the good things I had read about Linux servers, but without experience I felt safer in buying into the concept of “network appliances”. Network appliances can be a positive investment. They allow you to get some security in place in a short time and on a reasonable budget. You should however, be aware of what you are buying.

> In reality, most of the appliances we've seen in the Test Center lately are just general-purpose Intel platforms running Linux and some sort of security application. Usually, the version of Linux you get is modified to make it more secure - but usually not in a way beyond what you could do yourself if you know Linux fairly well. In other words, what you usually get in a security appliance is a Linux application bundled with hardware. (Rash, Wayne)

The first "internet appliance" I purchased was an "all in one solution" that was marketed to Windows users and small businesses without a networking staff. It was built on a secured Linux kernel and was configurable through a web browser. Patches and fixes were applied with a Graphical User Interface (GUI), thus avoiding the learning curve that comes with trying to use the command line. I felt this was a perfect solution for me and it worked pretty well for the most part, but managing your security in this way does have its disadvantages. With an appliance you are dependant on the vendor for security updates. When a new vulnerability for a common network protocol or service appears, the Linux community responds to it quickly and patches are readily available.

> If a particular vulnerability is identified in the core OS or related applications, programmers worldwide start simultaneously working to provide a patch to correct it. Some people do this for the common good, while others wish to gain some limelight for being the first one to fix the vulnerability. In either case, all Linux users benefit from these efforts. (Grimalia, Michael)

It can take weeks for a vendor to release a patch for your product. I monitored the vendors' support forum and watched as the Windows users like myself waited in frustration while the Linux-savvy users reported that they were downloading, compiling, and applying the updates from the command line. It is an uneasy feeling to know that your network is vulnerable because you are dependant on a vendor. In my opinion this situation is unacceptable and is what made me realize that knowledge of the Linux operating system is a valuable asset (if not a necessity) for network administrators.

Building a Linux-based perimeter network.

Step 1: Design.

Network design is a complex topic, which goes far beyond the scope of this discussion. You should study and carefully plan the topology of your network perimeter using the principles of defense in depth. The SANS reading room (http://www.sans.org/rr/) is a good place to start with many good papers applicable to your perimeter design. There is more information about Linux, DMZ structure, and defense in depth. You can also visit the CERT coordination center website, a valuable resource maintained by the Carnegie Mellon Software Engineering Institute. They have prepared a set of "Security Improvement Modules" which are free for the public to view, one of which contains very good coverage of firewall/perimeter topology (http://www.cert.org/security-improvement/practices/p053.html). Another source that must be mentioned is a book I purchased at a SANS conference, titled "Inside Network Perimeter

Security" (Northcutt, Stephen et al). It is truly the "definitive guide" and much of what I have learned about perimeter devices and their role in network security has come from this text. I highly recommend you buy and read this book.

Generally, the security devices found at the perimeter include static packet filters, firewalls, virtual private network (VPN) systems, intrusion detection systems (IDS), and screened subnets created by or between devices. Each device has a specific role and placement in your design.

The packet filter is the first or outer-most layer of your network security and is technically the most basic type of firewall. It is usually a router that lets you filter out specific types of traffic based on protocols or specific ports. It is not stateful, meaning that rules apply to both incoming and outgoing traffic. For example, if you need to use HTTP, port 80 has to be open on your packet filter and traffic can flow in both directions. Inbound HTTP would have to be secured by other devices in your topology. You can filter your incoming traffic (ingress filtering) for security threats like those listed on the SANS/FBI Top Twenty (http://www.sans.org/top20/) and you can filter outgoing traffic (egress filtering) to be sure that the only packets that leave your network are those that truly originate from your network. Proper egress filtering can prevent your systems from participating in denial of service attacks against others. You can also filter ports that are commonly used by Trojan horse programs. I will list a few of them as an example:

Trojan	**TCP-Port number**
NetBus 1.x	12346
NetBus Pro	20034
BackOrifice	31337(ELEET)
SubSeven	1243
NetSphere	30100
IcqTrojan	4950

These are just a few of the ports you could filter. If you go to www.google.com and search on "common Trojan ports" you will see many different lists and a great many ports that can be filtered for Trojans. There are so many that you could easily find yourself in a situation where you are "chasing your tail". You could spend your life updating the packet filter as Trojan writers change ports and create new Trojans. Filtering for commonly used Trojans is useful, but I feel the most effective way to fight Trojans is with a good anti-virus solution on the desktop and at the gateway. Stop them before they are ever installed on your systems.

The static packet filter can also be very helpful when your network is under a denial of service attack. If you can determine that a specific port is being used by an attacker, you can simply configure your packet filter to block all traffic to that port. Steve Gibson of www.grc.com experienced a Distributed Reflection Denial of Service attack against his website in May, 2001. He relates this experience in great detail on his website (http://grc.com/dos/grcdos.htm). Steve has a unique writing style (which has earned him both fans and critics) and the

page referenced is a long read, however I feel it is an excellent example of how a packet filter on a border router can “save the day” when your network is under certain kinds of attacks. Perhaps the most effective rule set for a packet filter is one that allows only the traffic you need to use (HTTP, SMTP, POP3, PPTP, etc) and denies everything else. You may have to experiment and re-open ports you didn’t realize you were using, but in the end you wouldn’t need to worry about Trojan ports or other vulnerabilities. The traffic that does come through can be managed and secured by your other devices. You could actually build a Linux router (www.linuxrouter.org) to use as your packet filter; however for this role and placement in the network, I feel a more “solid state” architecture is more stable and reliable. A router is diskless and is not prone to the issues of pc hardware. This is the only area of perimeter security where I feel the Linux solution available wouldn’t have the features and reliability needed for the required function. Linux routers could however, be an inexpensive and effective way to provide routing services on your internal network.

The next layer of security is the Firewall. Most networks today are protected with one or more of the three basic types of firewalls: packet filters, stateful packet inspection firewalls, and application proxy firewalls. There are pros and cons for each type of firewall and each has its place in your network. We have discussed the packet filter and its placement. The more advanced firewalls allow you to have different rules for incoming and outgoing traffic. For example, you can allow someone inside to connect to an outside source using HTTP but connections originating outside your network are denied or routed to a web server in your DMZ. The stateful inspection firewall is more advanced in that it “examines the contents of packets rather than just filtering them; that is, to consider their contents as well as their addresses.” (SupportNET) A stateful firewall would be placed behind the packet filter and act as a gateway to your network (if you choose to use only one firewall), or to a sub-network between your internal network and the outside world. An application proxy firewall is the most advanced type. It establishes a proxy relationship between your internal hosts and external hosts, preventing direct connections between these hosts.

> Each computer communicates with the other by passing all network traffic through the proxy program. The proxy program evaluates data sent from the client and decides which to pass on and which to drop. Communications between the client and server occur as though the proxy wasn't there, with the proxy mimicking the client when talking with the server, and the server when talking with the client. (SupportNET)

The nature of this proxy connection requires this type of firewall to be placed at the gateway to your internal network (if you choose to use application proxy). As complexity of the firewall types increases the latency increases. Your firewall choices will have to depend on your specific needs for your situation. A highly secure architecture would no doubt employ all three types of firewall: A packet filter at the border that only looks at the packet headers; a stateful firewall that examines the content and state of the packets; a screened subnet between firewalls; and finally an application proxy firewall that studies every packet of every session for every service.

The DMZ (demilitarized zone) is a screened/secured subnet between your internal network and the internet. It can be created between two firewalls as described above, or it could be accomplished with a single firewall that has three separate interfaces. The DMZ is home to web services that would be unsafe to place on the internal network such as a company's web site, email, ftp, or extranet server. Your firewall(s) give you control over what traffic goes into and out of your DMZ from both networks.

Everything that "lives" in your DMZ, the application servers and the firewall(s) that define its borders, can be created on a Linux platform. Just as many hardware vendors are using pc hardware to build "Linux security appliances" you can build your own Linux firewall. A good example of this is the Mandrake Linux "Multi Network Firewall". It has a list of features that rival any of the firewall appliances, and in my experience it is easy to install and configure. A complete feature list with screen shots can be found at the following URL: http://www.mandrakesoft.com/products/mnf/features . An important point to consider; depending on your security needs, you may prefer to run a diskless hardware firewall and save Linux for the application servers. Much can be done to harden the operating system of your firewall, but the firewall would still be weakened by any vulnerability of the OS. Hardware firewalls usually run a proprietary OS on solid state equipment which makes them more resilient because there are fewer moving parts. This is a matter of preference. I feel that for many people a hardened server running a Linux firewall is a perfectly secure solution. Consider your risk and the consequences of down time and choose accordingly.

Application servers can be any distribution you choose and can run any open source application you choose to handle the needed function, such as Apache (www.apache.org) for HTTP or Web Servers, or Sendmail (www.sendmail.org) for your mail server. Many of these applications will often come bundled in the Linux distribution you choose. There are also open-source security tools you can deploy in your DMZ to enhance the security of your server and your network design, such as Snort(www.snort.org) and Tripwire (www.tripwire.org) intrusion detection systems. Snort is a network based intrusion detection system that is "capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more." (Caswell, Brian) Tripwire is a host based intrusion detection system. It is designed to monitor your systems for changes that might occur if your system is compromised and notify you (the administrator) when they do. The design of Tripwire is very effective because it follows the principle of defense in depth. For example, it would be theoretically possible to hack the installation and tell it to report nothing, however every step involved in doing this (creating a new user, elevating privileges, editing or replacing files) would trigger an alert. Tripwire can see that someone is coming for it and can let you know before they get there. (Tripwire Inc.) That makes Tripwire a secure and valuable tool for protecting the applications in your

screened subnet or anywhere else. Tripwire for Linux is still open source but you can also purchase it for use on other operating systems and even network devices. You can learn more about this at www.tripwire.com.

Step 2: Choosing a distribution.

There are many Linux distributions ("distros") to choose from. You should do some research and decide which is best for you and your application. A good place to start your research is http://www.linux.org/dist/, a web site where you can search among the many available distros based on your needs.

I am partial to Mandrake Linux for several reasons. The Mandrake distributions are an easier starting point for those who are new to Linux ("newbies"). I have tried some distributions where I had problems installing drivers for the hardware I was using, but with Mandrake, the install process works as well as (if not better than) any Windows "plug and play" install I have ever done. Once it is installed, the GUI is well developed and robust, to the point that you almost don't need the command line at all. They offer complete solutions from desktop office suites to firewalls and mail servers. I highly recommend that you visit the following web sites: http://www.mandrakesoft.com as well as http://www.mandrakelinux.com/en/ to browse their product offerings, their philosophy, and their community pages to see which products will meet your needs.

"Plug and play" installations and helpful interfaces are nice but that is not all you need to look for in a Linux distribution. We are, after all, most concerned with the stability and security of the OS we intend to deploy. Mandrake distributions have also been certified by the Linux Standards Base, which was established to ensure stability and interoperability between systems and applications. This is an important credential to consider when choosing your "distro". You can go to http://www.freestandards.org/ to learn more about the LSB and to see current listings of distributions that have achieved LSB certification. Another good reason to use Mandrake Linux is that it is compatible with the "Bastille Linux" suite of security hardening scripts (www.bastille-linux.org). These scripts automate the process of securing your server against attack. They also explain and educate you about the steps involved as you go through them. This helps administrators to make informed choices based on their needs and makes it a much more informative process than just applying a patch or running a utility.

Step 3: Hardware.

One of the great things about setting up your first Linux network is how easy it is to get suitable hardware. You can simply use old hardware that comes available as you replace desktops (to meet the ever increasing demands of your Windows applications) and use it to build your own stable and secure Linux appliances. Dell and other vendors are now offering hardware that is certified as compatible with Linux. You can also find rack mount server class equipment from companies like www.penguincomputing.com. There are many commercial sources of hardware, but you may have all you need sitting in your storage closet

gathering dust. Linux is famous for the ability to run well on hardware that would be restrictive or prohibitive for other platforms. I have had a lot of luck with just pulling an old box off the shelf and running with it, but as a best practice you should take an inventory snapshot of your target hardware and compare it to the hardware compatibility list associated with your chosen distribution. There are software tools that you can use to inventory a system, one such tool is the free "Belarc Advisor" from www.belarc.com . If you download and run it on an old Windows box it will thoroughly analyze the hardware and display results in a browser window. You now have the data you need to search the HCL lists or look for Linux drivers if you have a problem later. As Linux matures, it is getting easier to find suitable hardware, and the distributions are becoming more adept at detection of components and installation of drivers. The HardDrake project (http://www.linux-mandrake.com/harddrake/) is one example of how the community is working to improve the ease of installation and hardware issues associated with Linux. Sponsored by Mandrake, volunteers have developed a GUI that anyone can download and that works with other distributions like Red Hat. I have done an install with HardDrake and it went very smoothly. The only issue I had was with a video card that was so new that it had not yet been included in HardDrake. I posted a question in an online user forum and within an hour someone had given me the correct driver nicely bundled in a script that installed it for me! Not so scary. I still have a lot to learn about using the command line, but that shouldn't deter anyone from taking these first steps and experimenting with Linux. As a complete "newbie" I have installed server, desktop, and firewall systems using Mandrake and it has been a good experience.

Step 4: Install, secure, test, and deploy.

You have designed your network, picked a distribution and found suitable hardware. Now it is time to put it all together and into place. If you have the luxury, it is a good idea to set up your new network in a lab environment. You could connect your systems and define the address scheme offline and separate from your existing network. In this way you could ensure that all of the devices function together and you can take steps to fine-tune the applications and secure them as much as possible before putting them into production. It may be possible, (if you have more than one public IP address) to run both your production and lab network in parallel, allowing you to connect and test against the live internet without impacting security on your existing network.

Go through the installation process for each of the machines you have chosen, ironing out any issues that occur until you have functional, stable systems. The choices you make during the install should match with the intended role for that system. Don't install unnecessary services. For example, if the system is not going to be a mail server, don't install Sendmail or any other mail related packages or services. If a package isn't installed, you don't have to be concerned with vulnerabilities to that package. If the vulnerability isn't there it can't be exploited. You should make sure the applications running on your systems are the latest versions and have current updates and patches. There

are a number of basic services that are installed by default such as echo, daytime, chargen, time, login, shell, and exec, which can be removed by commenting out lines in the inetd configuration file. A good discussion of unnecessary services running on server and stand alone Linux systems, including step by step instructions for editing the inetd configuration file can be found on the following University of California web page: http://www.ics.uci.edu/computing/unix/linux/services.php.

If you have decided to implement intrusion detection in your perimeter network, now is the time to consider the placement, install, and configure the alert settings. For example, you might want to run SNORT on one DMZ system and place Tripwire on any or all of the others. SNORT will allow you to monitor the activity. Tripwire can alert you if a hacker strikes and help restore damaged files.

If you are in a lab environment it is a good idea to put some test systems on either end of your network and simulate common network activity to see how it performs. This is also a good opportunity to run some security scans to see what vulnerabilities you can find. You can be certain that others will be testing your security on a daily basis - it is better for you to find any flaws in your design before they do. An excellent tool for this security assessment is Nmap, a free tool available for download from http://www.insecure.org/nmap/ and is also included with many Linux distributions.

> Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. (Insecure.org)

If you have left something open it is likely that Nmap will find it. Then you can address closing the ports or patching vulnerabilities. There are other ways you can easily and periodically scan your network. There are free and pay scanning services on the internet. Steve Gibson offers “ShieldsUP!” on his website, www.grc.com. It is a simple port scan and I have found it useful when experimenting with firewall rules or updates to quickly check the “stealth” status of the firewall. Another free scan site is www.pcflank.com. The pcflank site has a lot of preset tests as well as an advanced port scanner that allows you to target specific ports. Security metrics (www.securitymetrics.com) offers reasonably priced pay services including perimeter checks and site assessments. These web based alternatives can be a good way to get started with perimeter security testing. Using a penetration testing service can be a good way to take some burden from the network admin, and can make sure the network is being scanned from an impartial viewpoint thus protecting yourself from your own human error. It never hurts to get a second opinion, and you can never be too careful with your network security.

If your network is up and running and you are satisfied with the results of your testing, it is time to put your network in place. I would do this on a weekend or after hours. Apply the appropriate IP addresses to the interfaces, take a deep breath and cable it in. Start testing: generate some traffic, do some net based

scans, dial up from an outside line and run Nmap against the firewall and monitor the output from your IDS. Do as much as possible to make yourself confident in your security. Then set up a schedule to test it regularly. It is a good idea to subscribe to security mailing lists to be aware of what is happening in the security field. Network security is an ongoing process that is never finished.

In Conclusion.

We have discussed the many ways that Linux can be applied to your network perimeter. It offers substantial savings in hardware and software costs and is extremely useful and flexible. There are many excellent open source applications available for Linux that allow you to populate the perimeter with valuable services. Security tools are available to harden the operating system and monitor the perimeter for signs of attack. The value of Linux to your network is clear; the only thing holding you back is your comfort level and learning curve. For Windows administrators that learning curve can be intimidating, but the process I have described here can be a good way to get started. There are of course, many reference books and training courses available to help you get up to speed. The next time resumes cross my desk I will be looking for applicants with Linux experience or certification. I am convinced that their skills will increase security while having a positive impact on the bottom line.

References.

Weiss, Todd R. "Is Linux on the Desktop Inevitable?" February 7, 2003 URL: http://www.pcworld.com/news/article/0,aid,109266,00.asp

Rash, Wayne "Less Appliance Reliance" Posted March 21, 2003 URL: http://www.infoworld.com/article/03/03/21/12secadvise_1.html

Grimalia, Michael Russell. "The Role of Bastille Linux in Information Security" February 18, 2002 URL: http://www.sans.org/rr/linux/bastille.php

Northcutt, Stephen. Zeltser, Lenny. Winters, Scott. Frederick, Sarah Kent. Ritchey, Ronald W. "Inside Network Perimeter Security" June 28, 2002. New Riders Publishing

Gibson, Steve. "The Strange Tale of the DENIAL OF SERVICE Attacks Against GRC.com" March 5, 2002. URL: http://grc.com/dos/grcdos.htm

SupportNet. "Three Basic Types of Firewalls" URL: http://supportnet.merit.edu/m-intsec/t-firewa/text/3kinds.html

CERT Coordination Center “Design the Firewall System” URL: http://www.cert.org/security-improvement/practices/p053.html November 21, 2002

Caswell, Brian. “What is Snort?” URL: http://www.snort.org/about.html Copyright 2002, 2003.

Tripwire Inc. “Frequently Asked Questions (no. 11)” URL: http://www.tripwire.com/products/servers/faqs.cfm 2003

Insecure.org “Introduction” URL: http://www.insecure.org/nmap/