Global Information Assurance Certification Paper

Content Filtering and Security for Your Home Network

Chad A. McKinney
February 10, 2005
GSEC Practical Assignment 1.4c
Option 2: Case Study in Information Security

Abstract

The purpose of this paper is to show an easy and inexpensive way to increase security on your home or small business network by implementing a firewall and an internet content filtering solution. The solution will allow the security administrator or a parent to be able monitor and control their employee's or children's access to the internet.

The information provided will guide you through some basic configuration of a Linux firewall setup, Squid proxy setup and Dan's Guardian content filtering solution. You will also be provided with some alternative solutions to installing a separate Linux firewall and proxy environment.

Background

If you have children, your job is to keep them safe and secure – especially from sexual predators. Some ways you might initially think of is through physical security, keeping the doors and window locked when you are sleeping. Another way may be through monitoring, while they play outside in the back yard or while they ride their bike in front of the house. As a parent in these times, you are aware of these dangers, because they are visible to you.

The internet has become the sexual predator's new "playground". They have the ability to remain anonymous, impersonate children to gain trust, and then they have access to thousands of other pedophiles who are more than willing to share information and new techniques.

Times have drastically changed from when I grew up in the 70's. In more and more families both parents are working, sometimes late hours, just to keep their heads above water. This usually means less time to monitor and control your children's internet surfing activities. Implementing a content filtering solution we help monitor and control your household's internet activity, while you deal with the other issues of your busy lifestyle.

Also, the reason for this implementation and paper is so that you don't have to go through the embarrassing conversation with your young child (daughter) to answer, "What was that girl doing Daddy?" after you finger flubbed a URL to a cartoon website.

Why do it?

The statistics are scary, 1 out of 5 children ages 10-17, have been sexually solicited online.[1] The majority of this activity (89%) is happening via online chat rooms and instant messaging.[1]

The effect of looking at pornography on children, while it is not conclusive that looking pornography will create a sexual deviant, provides some areas for concern:

> "In a study of convicted child molesters, 77 percent of those who molested boys and 87 percent of those who molested girls admitted to the habitual use of pornography in the commission of their crimes."[2]

Another website provides an alarming statistic that 90% of 8-16 year olds

have viewed porn online – most while doing their homework.[3] I'm sure we all know from experience that doing an innocent web search can provide some disturbing results.

Internet porn is just one aspect of a content filtering solution. The admin/parent should be able to easily control access to gambling sites, racial hate sites, bomb making, and terrorist/extremist sites.

Before

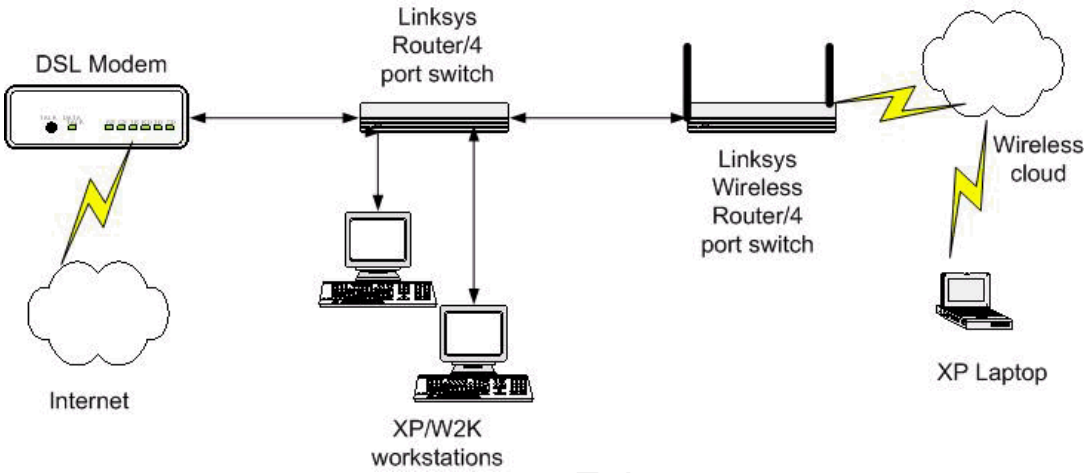

Figure 1 – before network diagram

The above drawing (figure 1) shows a basic broad band network. Current security is provided by a 4 port Linksys router/switch which sits just on the other side of the DSL modem. I ran a scan of my network to verify that the network is locked down from outside access and passes external port scans. The Shields Up[4] port scan came back with no response to probes detected.

The network is used for basic web browsing and email. I'm running Microsoft Windows OS on the client workstations. The only security precautions are to do patch management by periodically connecting to Microsoft to retrieve and updates/patches. Outlook client and Internet Explorer (IE) browser have been configured for basic security measures. I am also running MacAfee virus scanning software on the workstations.

The Outlook client has some default features turned off. The preview pane is disabled in all folder views. This stops Outlook from auto opening any email which helps to prevent installation of spyware or displaying inappropriate content.

Problems

- All workstations are internet aware, this poses a problem only if human interaction launches an executable off of a website or email attachment.
- There is little to no logging of activity
- The internet is wide open, no controls on where you can go or what you can see.

During

My main objectives in coming up with a content filtering solution, were to minimize cost (free), solution should be easy to use, and solution should not be time consuming to implement. I also had some future enhancements, installing a file/web server, that I wanted to prepare for which factored into my decision.

I decided to go with a Linux firewall, Fedora core 3, running Squid proxy server and Dan's Guardian. It's debatable whether this met my objectives of easy to use and not time consuming to implement, but this solution offered the greatest flexibility and positioned me for my future enhancements.

I initially thought about running software or specialized browsers to control access to the internet. The next section covers some brief analysis I performed of alternate web content filtering solutions.

Analysis

The quickest and easiest way to control where kids are going is to only allow them to use a kid friendly search engine. These search engines only return results of websites that have been checked by a human and have been deemed kid age appropriate. [5] Testing of these products proved to be effective in limiting the amount of information returned on a search to kid appropriate content.

Google[6], Yahoo[7] and other well know search engines offer SafeSearch Filtering option. This is accessible via the preferences link off of the main search page. Both products claim to block explicit adult content words and pictures. Testing of these features proved to be less than desirable. A simple search of "porn" returned countless entries that contained explicit text in the description and also allowed access to the site.

I've tried the "content" settings in previous versions of Microsoft Internet Explorer and it seemed to me that I got a lot of false positives. The other down fall of trying to setup web content filtering through the IE content settings, is that I would have to manage it individually on each system.

The final possibility was to pay for and install third party software that specializes in content filtering and monitoring.

Linux installation

This was my first attempt at installing Fedora Linux. I've installed other versions of Linux, SUSE and Mandrake, in a workstation type configuration, so my wife could use them for college classes. So, this is my disclaimer, I'm a "nube" to Fedora Linux, I am only providing the steps I had to do to overcome my configuration challenges. I highly recommend bookmarking the following website, it was a great help in setting up my Linux system: www.linuxhomenetworking.com [8]

First I had to obtain the software, Fedora Core 3 can be found off of the Fedora RedHat web site.[9] I downloaded the 4 ISO images and then used burning software to burn them to CDs.

My next task was to get Fedora Core 3 Linux running on an old workstation. I had a couple systems to choose from, a Pentium III 1Ghz machine and a Pentium II 400Mhz machine. I naturally started with what I thought was the better machine – the PIII 1 Ghz. This ended up being a big mistake. I ran into nothing but problems – especially with video. The PIII, while a faster processor, lacked a sufficient video card, which was built onboard. The PII has an ad-on AGP 32mb video card. The PII had previously been running Windows 2000 (W2K), so I ran into some problems with the initial installation attempt. The install was hanging when I attempted to do the Fedora media check. I ended up having to reset my BIOS back to factory defaults and replace the CD-rom drive. It was also recommended to me by several individuals and websites to make sure that the system was not directly connected to the internet. My system was safe because I was connecting it to the Linksys router, which passed the external vulnerability scan.

The hardware issues resolved, we can move onto the software installation. I went with the more secure installation, limiting the amount and type of packages I installed to just what I planned on using. This was an attempt to follow the recommendation from the SANS courseware, "If you don't need it, turn it off…"[10] My thought was "If you don't need it, don't install it", while I consider this a good philosophy it ended up coming back to bite me when I tried to install a package from source files and didn't have any of the compilers. So, make sure you get the developer's packages.

The plan was for this Linux box to be used as a Firewall. In order for it to function as a firewall I needed two interfaces. One interface will be my external interface, or untrusted (eth0), which faces the internet. The other interface is my internal interface, or trusted (eth1), which faces my protected workstations. Fedora has a utility similar to plug-n-play for Linux called "kudzu" which does a scan of your system for hardware changes. The program successfully detected and installed my second network card (figure 2).

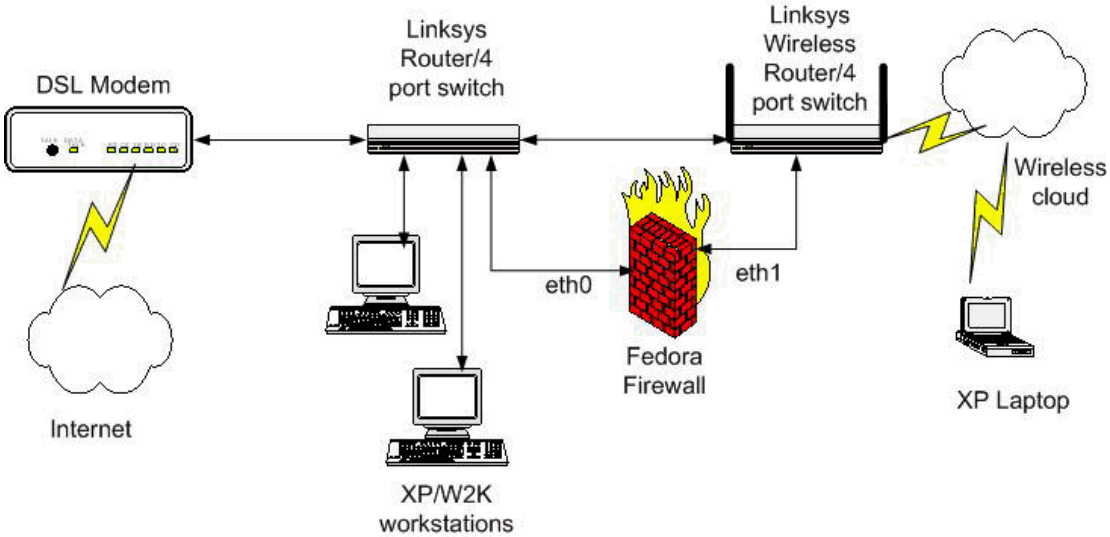

Figure 2. – network diagram during firewall load

Services

Now that everything is installed, the system needed to be locked down. The SANS courseware offers some good basic steps to look out for in stopping your services.[10] The following command will allow you to see a list of services installed on your system and determine the appropriate run levels:

chkconfig –list

I think this is a good way to see what is installed and running, but I found that with my limited knowledge I didn't know by the abbreviated names what the services were. I found that running the following command offered similar information and provided some detail about what that service does:

#system-config-services

Be sure to pay attention to the run level that is being displayed, you'll have to click on the service and then modify each run level for that service. Run levels are where each service will run in the boot sequence. Typically, if you want something to run, you need level 3 (command prompt) and level 5 (X-windows) "ON". Turn off all the run levels for a service if you do not want it to run.

Updates

Since I had a network connection to the internet through my Linksys router (see figure 2 above), I felt comfortable connecting an unpatched Linux system to my network. Before I moved the firewall out to the internet, I wanted to make sure that I wasn't vulnerable to any hacks. Fedora core 3 comes with a handy utility, up2date. I'm running the KDE desktop and this shows up in the tray as an exclamation mark when you need to update your files. I made sure that I updated all of the necessary files prior to moving the system on the other side of my router. Another utility for updating your system files is "apt-get". From a terminal prompt you can type the following commands, one to get the "apt-get" program and the other to update your distribution:

rpm -Uvh http://atrpms.net/dist/fc3/atrpms-kickstart/atrpms-kickstart-25-1.rhfc3.at.i386.rpm

Once you have apt-get type the following command:

apt-get dist-upgrade

This will go out and get all of the packages and install them for you – dependencies and all.

Network configuration

In order to remove some complexity and routing issues with my network, I decided to do away with one of my Linksys routers – the 4-port wired and stay with the wireless router/switch (figure 3 below). My old configuration was setup using 10. addresses and I opted to change this to the 192.168.x.x address space. Since my firewall's internal interface was now going to be my default gateway for all of my workstations I had to setup a DHCP (dynamic host control protocol) server on my firewall and a DNS (domain name system) server so the workstations could resolve IP addresses for mail and VPN (virtual private network).

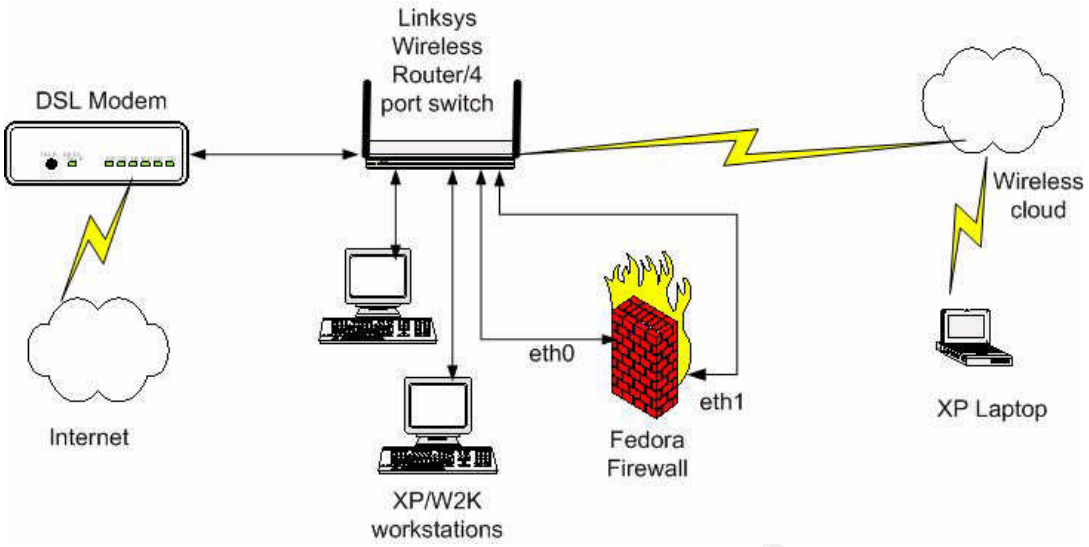

Figure 3. – simplified network diagram (still configuring firewall)

The DHCP server was pretty straight forward for configuring. The below location contains an example of how to setup your DHCP server: [11]

/usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample

Depending on what you installed when you loaded the Linux OS dhcpd may not even be installed. You can use apt-get to pull down the latest version of dhcpd:

apt-get install dhcp

Output of DHCPD.conf:

```
# dhcpd.conf
#
# Configuration file for ISC dhcpd (see 'man dhcpd.conf')
#
ddns-update-style ad-hoc;

subnet 192.168.xx.0 netmask 255.255.255.0 {
        range 192.168.xx.100 192.168.xx.110;
        default-lease-time 86400;
        max-lease-time 86400;
        option subnet-mask 255.255.255.0;
        option broadcast-address 192.168.xx.255;
        option routers 192.168.xx.10;
        option domain-name-servers 192.168.xx.10;
        }
```

I had some more difficulty in installing the DNS server. DNS wasn't installed so I used apt-get to get the files:

apt-get install bind

Everything appeared to install properly, but when I went to run DNS it errored out. I'm not sure why this happened but my zone files did not get created in /var/named. I was missing named.ca, named.local, and localhost.zone. Rebuilding these files corrected my DNS issues. [12]

IPTables/Firewall

I initially tried to build my IPTables by using a GUI based firewall builder – FWBUILDER. [13] It was a very easy to use product and had a look and feel of some other commercial based firewall products. I was unable to get it to successfully control the traffic the way I wanted. There are other firewall creation packages out there and Linux has the built in Lokkit utility for configuring your firewall. I unfortunately did not have a lot of time to try out several alternatives. So, I ended up contacting a friend who was very knowledgeable about IPTables. [14] He was able to break IPTables down into a few key points:

1. Input chains processes traffic that is destined for the firewall.
2. Output chains processes traffic that is originating from the firewall.
3. Forward chains processes traffic that must go through the firewall.
4. Pre-Routing chain performs NAT (network address translation) for destination addresses.
5. Post-Routing chain performs NAT for source addresses.

I created a script based on the information that I received from him and additional sources on the internet. [15] See the referenced links for additional information on iptables. I highly recommend starting from some examples or modify a friend's script to match your network configuration.

In order to get my firewall up and running through this phase of the configuration, I needed everything from the outside blocked and everything from the inside allowed. I setup the firewall for stateful packet inspection, which means that traffic that I initiated from the inside is put into a state table and that table is checked when the traffic returns from the outside. If a request is not in the state table, then it is not allowed. After this phase of the iptables configuration was completed I was able to get out to the internet.

I next started to secure the firewall down by adding Anti-spoofing rules to my input chain. Anti-spoofing prevents someone from the outside, entering on eth0, from coming to your firewall and acting or spoofing like they are part of your network by having your same internal network address range. I also changed my wide open input and forward chains and only allowed traffic from my internal network space. Next I enabled some additional features like "TCP SYN cookie protection" and "log_martians". The syn cookie protection protects against DOS (denial of service) attacks and the log_martians looks for incorrectly formed IP addresses. Finally, I disabled "ICMP redirect acceptance" and "accept_source_route" or source routed packets.

Now I was ready to move my firewall on the other side of my Linksys router – directly connected to the internet (figure 4 below). This went smoothly, traffic is still being allowed from my internal network and packets that weren't

initiated by me are being dropped by the firewall and being logged in /var/log/messages.

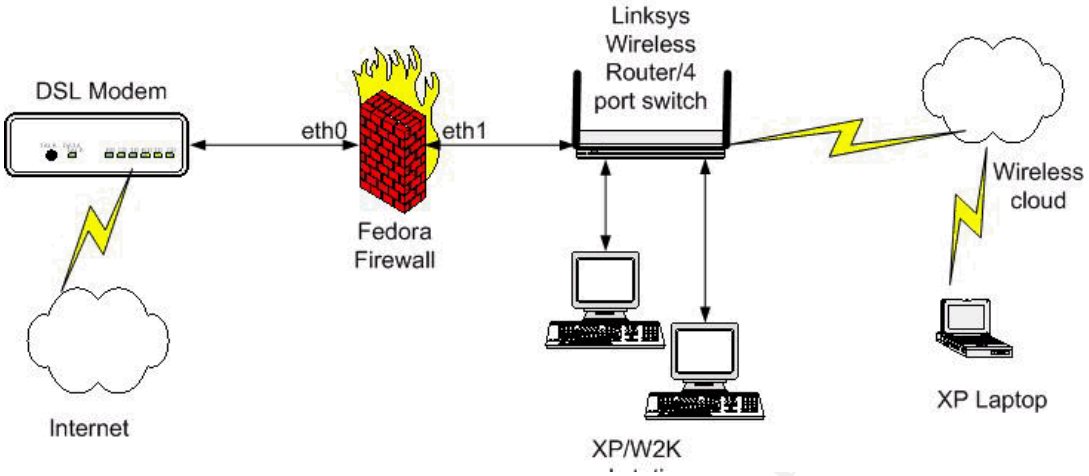

Figure 4. – final network configuration diagram

Squid Proxy [16]

Now the traffic is successfully flowing through the stateful packet inspection firewall, I can move onto the Squid proxy server installation. A proxy server is a device which handles caching or storing of internet related traffic. Once a page has been cached, subsequent requests for the same page are pulled from cache rather than requested from the internet. This process saves time and bandwidth. Most good websites add tags or information to the HTTP headers to help the proxy server determine whether or not to cache the site and for how long. If the time has not expired on a cached object, it will not attempt to go out to the internet to get a fresh copy.

Fortunately, Squid was one of the packages that I had chosen to install when I loaded the Linux OS. I ran chkconfig –list to verify that squid was running at the appropriate run levels, which it was. The main configuration file squid.conf is located in /etc/squid. Squid, by default listens on port 3128 on all available networks. I had to un-comment the http_port in order for squid to listen for traffic on this port. The visible_hostname option needed to be set, which I just set to Home_proxy. I added an http_access statement to allow “home_network”. Next, I needed to add an acl (access control list) entry to allow “home_network” which pointed to my internal network address space. At this point, Squid is now configured to allow caching of internet traffic.

I had to go to my workstations and configure the Internet Explorer browser proxy settings:

Tool -> Internet Options -> Connections -> LAN settings

I entered the IP address of the inside interface of my firewall and port 3128. Access to the internet now appeared to be going through the Squid proxy server. Well, I decided to verify that this was the case and changed the port from 3128 to 3130. Squid should not be listening on this port so my next request should get

blocked. Unfortunately, it did not get blocked because I had forgotten to adjust my iptables script to limit traffic from my internal network down to just port 3128. After correcting my firewall (iptables) script, I was not able to get out to the internet on any other ports other than 3128.
Modified iptables line:

```
/sbin/iptables -A INPUT -i eth1 -p tcp --dport 3128 -s
192.168.xx.0/24 -j ACCEPT -m state --state NEW
```

Dan's Guardian [17]

Dan's Guardian filters traffic based on multiple methods. "These methods include URL and domain filtering, content phrase filtering, PICS filtering, MIME filtering, file extension filtering, POST limiting. " [18]. I initially attempted to download and install DansGuardian via the source code. This, as I mentioned above in the Linux installation section, did not work out as directed. I was missing several key compilers and other necessary files. I was able to get some of the files by running the following command to retrieve the developer's files:

yum groupinstall "Development Tools"

This command retrieved the compilers, but I still ran into trouble completing the installation and ended up aborting the attempt with the source files. The only rpm files available were listed for Fedora Core 2, which was the reason I went with the source files, but I was able to successfully install it on my version of Fedora. Now that DansGuardian (DG) was installed and running, I had to re-configure my firewall script to listen on port 8080, which is the default port for DG.
Modified iptables line:

```
/sbin/iptables -A INPUT -i eth1 -p tcp --dport 8080 -s
192.168.xx.0/24 -j ACCEPT -m state --state NEW
```

The traffic is now flowing from the workstation, to Dan's Guardian on port 8080. DG does the web content filtering and if it is allowed passes the request off to Squid on port 3128. Squid checks to see if the page has already been cached, if so it serves up the content back to the requestor. If not, Squid requests the page from the internet, caches the page locally and then serves up the content to the requestor. If DG deems that the request falls into a blocked category, then DG serves up a blocked message to the requestor, detailing the reason for the block.

One more trip back to the workstations. First I wanted to make sure that the workstations could no longer get on the internet, because they were still pointing at the old proxy port of 3128. This proved to be successful; the old port was no longer allowed. I changed the proxy port to 8080 and I was able to surf the net again. Now for the ultimate test, I pulled up Google and typed in "porn". I immediately received back a blocked message "Banned Regular Expression URL found". My wife and I proceeded to try and break DG and we were

unsuccessful.

Now that everything was working as designed, I decided to try and make sure that everything was locked down the way I intended for it to be used. I ran a "netstat –an |grep LISTEN" to see what ports were listening for traffic and the interfaces they were listening on. The risk in having a service listen on an external interface is that it provides a hole back into my network, which may provide a hacker all the access he needs.

Output of netstat before config changes:

```
tcp        0      0 0.0.0.0:8080            0.0.0.0:*     LISTEN
tcp        0      0 192.168.xx.xx:53        0.0.0.0:*     LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*     LISTEN
tcp        0      0 0.0.0.0:631             0.0.0.0:*     LISTEN
tcp        0      0 0.0.0.0:3128            0.0.0.0:*     LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*     LISTEN
```

I expected to find Squid and Dansguardian listening on both interfaces, but I also found that port 631 was listening on both interfaces. A little research and I found that I had missed a service "cups" which is a UNIX printing service. I disabled the cups service and move on to locking down Squid and DG to listen only on the internal interface. DG /etc/dansguardian/dansguardian.conf had an option called "filterip" that in default configuration was blank, which meant it would listen on all interfaces. I added the internal interface IP address of my firewall and this removed the ability for DG to listen on the outside interface. Squid's /etc/squid/squid.conf needed to have the "http_port" option changed. By default, the configuration only lists the port number, in this case 3128. I initially changed the config to the internal interface IP address of my firewall with port 3128, as I did with DG, but it was still listening on the outside interface. I ended up adding 127.0.0.1:3128 to the config and this corrected all the issues.

Output of netstat after config changes:

```
tcp        0      0 192.168.xx.xx:8080      0.0.0.0:*     LISTEN
tcp        0      0 192.168.xx.xx:53        0.0.0.0:*     LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*     LISTEN
tcp        0      0 127.0.0.1:3128          0.0.0.0:*     LISTEN
tcp        0      0 192.168.xx.xx:3128      0.0.0.0:*     LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*     LISTEN
```

After

I needed to verify that all these changes to my network did not make me vulnerable to attack. I had a friend run an nmap scan of my system and everything came back as filtered.

```
Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2005-02-21 09:43 CST
Initiating SYN Stealth Scan against atlnga1-arx-x-xx-xxx-xxx.atlnga1.dsl-verizon.net (x.xx.xxx.xx)
[1663 ports] at 09:43
SYN Stealth Scan Timing: About 28.50% done; ETC: 09:44 (0:01:15 remaining)
```

```
The SYN Stealth Scan took 103.82s to scan 1663 total ports.
Host atlnga1-arx-x-xx-xxx-xxx.atlnga1.dsl-verizon.net (x.xx.xxx.xx) appears to be up ... good.
All 1663 scanned ports on atlnga1-arx-x-xx-xxx-xxx.atlnga1.dsl-verizon.net (x.xx.xxx.xx) are:
filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 104.190 seconds
```

You may ask, 'what kind of security did you enhance?', since the original scan showed "no response to probes detected". As the administrator of this network I now have complete control over who can access my firewall and with what type of traffic. I have accurate and detailed logging of incoming and outgoing traffic. I have complete control over the type of content that I am allowing my family to view while surfing the net. I can block files from being downloaded that may contain executable code such as .exe, .com, or .zip.

Basically, I enhanced security from the inside out. Before and even now the network was/is secure from the outside. The enhancements I've made to my network have now reduced, if not eliminated, the risk of an internal user downloading malicious code from the net and running it on a workstation. Even if a piece of code is downloaded to a workstation, the workstation does not have a direct route to get to the outside.

Enhancing security to a locked down level hasn't been without its share of challenges. For example, while working on this paper and transferring the files between my work computer and my home computer, I found that I could not download my .doc file from "gmail". I also found a clothing catalog website was being blocked for a .vbs extension. These are easy things to correct, but something to be aware of when implementing this in your home network.

Conclusion

Whether you are working in a major Corporation or just trying to keep your home network clean and safe, you can't deny the need for Web Content Filtering. As a Corporate security administrator, you must also factor in the legal ramifications of not keeping a clean, non-hostile, work environment.

A couple of weekends and a little research and you can enhance your home network security, give yourself piece of mind, and enhance your children's internet learning experience.

References:

1. ProtectKids.com 15 February 2005. <http://www.protectkids.com/dangers/onlinepred.htm>
2. ProtectKids.com 15 February 2005. <http://www.protectkids.com/effects/harms.htm>
3. TopTenReviews website 15 February 2005. <http://internet-filter-review.toptenreviews.com/internet-pornography-statistics.html>
4. Shields UP Home Page 8 February 2005. <https://grc.com/x/ne.dll?bh0bkyd2>
5. Sullivan, Danny. "Kids Search Engines." SearchEngineWatch. 25 February 2004. 7 February 2005. <http://searchenginewatch.com/links/article.php/2156191#guides>
6. Google website preferences page 16 February 2005. <http://www.google.com/help/customize.html#safe>
7. Yahoo website preferences page 16 February 2005. <http://search.yahoo.com/search/preferences?pref_done=http%3A%2F%2Fwww.yahoo.com&fr=fp-top>
8. Harrison, Peter. Home page. 19 February 2005 <http://www.linuxhomenetworking.com>
9. Fedora Home page. 17 February 2005 <http://fedora.redhat.com/download>
10. SANS Institute. Track 1 SANS Security Essentials Volume 1.6. SANS Press, Jan 2004
11. Harrison, Peter. "Chapter 8: Configuring the DHCP Server". LinuxHomeNetworking.com. 19 February 2005. <http://www.linuxhomenetworking.com/linux-hn/dchp.htm>
12. Harrison, Peter. "Chapter 18: Configuring DNS". 19 February 2005 < http://www.linuxhomenetworking.com/linux-hn/dns-static.htm>
13. Firewall Builder Home page. 2003. 19 February 2005 < http://www.fwbuilder.org/>
14. Rupprecht, Eric. "Case Study of IPTables and Freeswan IPSEC" 10 November 2003.
15. Harrison, Peter. "Chapter 14: Linux Firewalls Using IPTables". 19 February 2005 <http://www.linuxhomenetworking.com/linux-hn/iptables-intro.htm>
16. Squid Web Proxy Cache Home page. 20 February 2005. <http://www.squid-cache.org/>
17. Barron, Daniel. Home page. 20 February 2005 <http://dansguardian.org/>
18. Dan's Guardian website <http://dansguardian.org/?page=introduction> 03 December 2003