Global Information Assurance Certification Paper

Desktop Firewalls in the Enterprise
GIAC Security Essentials Certification Version 1.4b - Option 2
Jeffrey Heller

Abstract

This paper is intended to show the need for desktop protection in a Local Area Network, even though the network is behind an existing firewall. This represents another layer in Defense in Depth. This paper will show the process of selecting a suitable solution for one enterprise, and will hopefully provide guidance to the reader for implementing his own solution. Snapshots of the enterprise will present three pictures of the network segment that we are covering. The initial snapshot shows the desktop computers after hardening. The during snapshot shows the process of evaluating and configuring our selections. The after snapshot shows the results we were trying to achieve.

An Analogy

Imagine that you are the dean of a small university in a small college town. Ten years ago, the campus was wide open, and anyone could come and go as he pleased. The town's main street and several side streets ran right through the center of the university. Doors to residence halls, the library, and other buildings were not locked. It was all open.

During the last ten years, the town grew rapidly. Also during this time, government and industry invested heavily in research programs at the university. Security became a concern. After some small crimes, fences and walls were put around the campus. Town streets were routed around the school. Gates were manned, students and faculty were required to carry badges, and visitors were always escorted - through the gates. Because of all of these precautions, everyone felt secure. Because of this and because of the inconvenience involved, the doors to classrooms, labs, and offices do not even have locks. This never seems to be an issue until a visitor walks away with some enormously expensive property. It is time to put some locks on the doors.

Every analogy falls apart if you carry it far enough. In our analogy above, the university is the campus network you administer. The town is the Internet. The gate is the perimeter firewall, and people are the packets of information - good and bad - that come and go. The doors to the classrooms, labs, and offices are the network cards to the desktop workstations. Where this analogy falls apart is in the fact that it would be unthinkable to leave those doors unlocked. Yet, in a Microsoft Windows network, there are no locks in place.

As the 1990's progressed, the typical desktop computer in business organizations evolved. What started out as an isolated, slow box became more powerful and fully connected to a network of other powerful computers. This

network connected to the Internet - other networks of other powerful computers. It seems hard to believe today, but ten years ago there were no firewalls between most organizations. As the Internet grew rapidly, government and industry invested heavily in computing resources and infrastructure. Scanning and Hacking tools were developed and put to use in unscrupulous ways. A few intrusions occurred on everyone's networks, and some high profile crimes made the news. Security became a concern. A firewall was placed on the perimeter of the network.

A perimeter firewall is really a gate between the Local Area Network and the Internet where packets of information are inspected to ensure that they are authorized to proceed from one side to the other. This inspection can be done by a computer using software, or by dedicated hardware. In either case, the firewall is operating under a set of "rules" which are written by the administrator.

Personal firewalls have existed for home and small office users for some time. Back when Windows 95/98 was the most prevalent operating system for desktops, an organization's users might install personal firewalls to "their" desktop personal computers. This had the effect of locking out administrative functions required by the people responsible for the computers. As a result, it would often become a corporate policy that personal firewalls would not be allowed on users' desktop computers. The exception might be for notebook computers that traveled with users and connected to the internet via dial-up.

Firewalls are not perfect, however. In simplest terms, a list of open ports that configure the network device can act as a firewall. This can be effective, because only traffic that matches the rule will pass to the other side. However, when common ports are open, exploits can be geared to use those ports.

If we assume that all malicious packets and traffic are on the 'outside' of the network, then we can add another feature to the firewall - statefulness. When the computer on the inside of a firewall requests communication with one on the outside of the firewall through an open port, the firewall is set up to expect and process the reply - even if the return port is not open to inbound traffic. A communication state exists.

Trojan Horse Programs are software that looks friendly, but contain malicious code. Trojans have become the cause for us not to assume that all malicious packets and traffic are on the outside of the firewall. So, another level of sophistication was added to firewall protection. This is called program control, or outbound filtering. The firewall examines the programs which are accessing the outside network, and compares them to a list it keeps. If a program is not on the list, or if the program has been altered by a virus or other cause, the firewall administrator is alerted. It does require the administrator to be aware of his programs, and ready to make a judgment call at a moments notice. This would seem to be an adequate control, but the added protection provided by outbound

filtering is entirely illusory.(1) This is because programs operating in the Microsoft Windows environment can hand control of their actions to other programs. This is usually transparent to the user. However, It should not be transparent to the firewall.

The next level of sophistication for firewalls is called component control. With this, the firewall examines new programs, trusted programs and trusted component programs (such as Windows .dll files) and their MD5 hash values calculated by the firewall. If any of these do not match the firewall's list of trust, then the firewall administrator is alerted. Compiling and maintaining these lists are parts of the administration. One can see that this could become an enormous time investment. With users and in-house programmers trying new programs, using different operating systems, and patches coming out regularly, the job would be unending. The firewall choice comes down to a cost versus benefit study. The differences may be extreme.

Firewall Types Comparison

Type:	**Initial Costs:**	**Ongoing Costs:**	**External Protection:**	**Internal Protection:**
No Desktop Firewall	Low	Low	Poor	Poor
Stateless Packet Filtering	Low	Low	Good	Fair/Good
Statefull Filtering Firewall	High	Low	Good	Fair/Good
Program Control Firewall	High	Moderate	Good	Fair/Good
Component Control Firewall	High	High	Good	Fair/Good

- Initial costs include purchasing the product.
- Ongoing costs include administrative upkeep and maintenance of the software.
- External protection ratings are derived from the results of various scans against the system.
- Internal protection is very much dependent on the administrator to provide maintenance of the software and policies which prevent internal threats from materializing.

Initial Snapshot

One motivation for this type of project comes about when your department is flagged by a scanning examination. A consulting team might come in a few times a year to shake things up a bit and prove to management that the network and systems still have vulnerabilities. Their tools of choice are Symantec NetRecon, Insecure.org's NMAP, Internet Security Systems' Internet Scanner, or Foundstone's Superscan. The figures shown in this report were produced by Symantec NetRecon. Other tools produced fairly similar results. The scanning machine is on a different subnet from the scanned machine, but both are on the same side of the perimeter firewall. They are both members of the same

Windows NT domain. If the scanning computer is not a member of the domain when the scan is done, the change in results is usually insignificant.

Scanning tools can work on specific computers, on groups of computers, or on a range of IP addresses. Figure 1 shows a typical scan against a Windows 2000 professional system with service pack 3 and all patches installed. It lists the level of vulnerability on a scale of 0 -100 according to the tool's developer. Next to that, it provides a short description of the vulnerability itself.

Risk	Vulnerability
95	Malformed RPC Request Can Cause Service Problems
75	Signed ActiveX controls marked safe for scripting in ...
48	Windows NT system caches logon credentials
48	base system objects not audited
48	password filter not enabled
48	LanManager authentication permitted
45	OS/2 subsystem enabled
45	POSIX subsystem enabled
45	guest account can access system event log
45	guest account can access security event log
45	event auditing failure permitted
44	auditing of rights not enabled
43	Windows NT page file not cleared at system shutdown
42	nbname service enabled
42	local users can install print drivers
42	file shares may be enumerated remotely
42	nbsession service enabled
37	unrestricted null session enumeration possible
37	guest account can access application event log
19	DCOM enabled
18	network access to CD-ROM possible
18	network access to floppy disk drive possible
17	open UDP port may allow unauthorized activity
17	open UDP port may allow unauthorized activity
16	network resource identified
16	network resource identified
16	username of last login displayed
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
16	network resource identified
14	open TCP port may allow unauthorized activity
10	IP name obtained
10	IP name obtained
5	IP address found from name
5	IP address found from name
3	logon dialog box allows system shutdown

Figure 1 Screenshot of typical desktop system scan

During Snapshot

Turning off services was not helping much. Patches, policies and prayers were all employed to harden the desktop machine. These were user oriented desktop

computers running Windows 2000. Each user has his or her own computer, and they are used for everything from word processing to program development. There are also community computers for document scanning and for multiple CD burning. While the standard installation of Windows 2000 does have many services that are not needed or used, everything that could be locked down had been. It is possible that the scanning tools were reporting false positives, but our segment was still getting tagged. With the complexity of today's operating systems, and a lack of direct administrative control of the network card, how could anyone be sure?

Today's scanning tools are only identifying today's known exploits. Patching and upgrading are reactive solutions to weaknesses which have been identified. Perhaps a more proactive solution could improve the situation.

After some discussion with a fellow administrator, the concept of using firewalls was brought up. Administrative needs allow for some flexibility in installing software. So, even if there was a corporate policy against firewall software, it might be obsolete. There were two problems that had to be overcome. First the firewall had to be controlled administratively - hopefully from a central location. The end user could not make rule changes that would defeat administrative processes. Second, it should not alert the end user as to either its presence, or its blocked packets. Logging was desirable, as long as the log files did not get too large. We would not be required to use the log files for auditing, so we were mainly interested in the aid in fine tuning the firewalls.

The Search

Most anti-virus vendors have already geared their anti-virus products to support the enterprise. The anti-virus software is installed from and controlled by a dedicated server. The end user cannot uninstall the software, and the configuration options are locked. The programs cannot be turned off easily by the user with the task manager. Virus definitions and upgrades are handled by the server automatically. The server is the only machine which has to make contact with the vendor. The programs running on the desktop are not resource hogs, so the machine stays stable. Could desktop firewall software act the same way?

Symantec Desktop Firewall

First up was Symantec Desktop Firewall. It is available from Symantec as part of their overall enterprise support packages. This was used as a proof of concept. When it was installed there was a configuration interface available to the user. This would have to be removed, and there appeared to be an administrative centrally controlled option, but we never got as far as testing that. The reason for this will be explained further down.

The default setup includes a rule set which blocks many outbound ports used by known trojans. The method for applying rules is this: the first rule that fits is processed. This is appreciated by anyone familiar with setting up routers. It is fully configurable, but the interface software seemed to use a lot of resources. The logging of blocked packets did not work with the few Windows XP Professional machines we have in use now. This was discovered to be a problem caused by the installation of Service Pack 1 for Windows XP. It breaks the logging function for other products as well. This could probably be survived until a new patch came out to correct that behavior. After setting up the options desired, the interface comes to a tab with a toggle to start the firewall when Windows starts. Of course, that would be selected. On the next restart, the machine or certain services would seem to hang or lock up. Uncheck the toggle, and restart, and all would be well. A quick look at Symantec support showed this to be a known problem - first on their list.(2)

With this product all but eliminated, we still wanted to try a scan. The results were impressive.

Risk	Vulnerability
16	network resource identified
16	network resource identified
10	IP name obtained
5	IP address found from name

Figure 2 Screenshot of scan with Symantec Desktop Firewall running

This showed us we were on the right path. Firewalls would protect desktop machines from certain scan exploits. In addition, with certain rules in place, file and printer sharing, and any administrative function could be performed on the machine. These included terminal services, and remote computer management.

Zone Labs Integrity Enterprise

Zone Labs, Inc. was known for their inexpensive home firewalls. Perhaps with a little programming, these could be configured to be administratively controlled. Looking at their website showed a different product - Integrity Enterprise.(3) It is an administrator controlled, configurable firewall product. It is available in a demo version for trial use, and was designed to support the enterprise. It requires a dedicated server, but when contact is lost with the server, the program defaults to a user mode. This would be useful for notebook computers that traveled away from the domain.

The configuration interface on the server takes some time to figure out, but parts of it will be familiar to anyone who has used Zone Alarm. That helps in the setup.

The default setup starts out with certain programs being given permission to cross the firewall. Component control is enabled by default. It also was impressive at blocking a typical scan.

Risk	Vulnerability
42	nbname service enabled
17	open UDP port may allow unauthorized activity
17	open UDP port may allow unauthorized activity
17	open UDP port may allow unauthorized activity
16	network resource identified
16	network resource identified
10	IP name obtained
5	IP address found from name

Figure 3 Screenshot of scan with Zone Labs Integrity client software running

Although it is centrally controlled, the end user gets the firewall alerts, up to a maximum of 500 during an intense scan. There may be a policy that disables this, but it was not found. The firewall configures the hash values for default components when the client software is installed from the server. An icon shows up in the user's taskbar that shows the software is installed. Instead of logging failed attempts to cross the firewall on the server, a graphical display on the server interface shows approximately how many attempts have occurred over a time period.

PktFilter

It was then that we discovered a utility for Windows 2000/XP computers named Pktfilter. It is available at: http://sourceforge.net/projects/pktfilter/. The author is Jean-Baptiste Marchand.

PktFilter is a service to control the IPv4 filtering driver in Windows. It is in beta release at this point, and is distributed under the BSD license. The source code is distributed as well as documentation, so improvements can be hammered out by those who are so inclined.

This is a Stateless Packet Filtering Utility which configures the IPv4 filtering driver. It functions strictly as an interface to the driver. This driver is normally configured by the Routing and Remote Access (RRAS) service, but the Pktfilter program is small, and dedicated to IP filtering. There may be a conflict with RRAS, according to the documentation, but no errors of this sort have been seen.

The program is configured through a text file usually named rules.txt, but another name could be specified during the installation. The rules grammar - as stated in the documentation - sometimes still produced a syntax error. This was usually as a result of the "any" argument for the protocol. This simply created the need

for separate rules. Logging of blocked packets is supported, and is written to a text file by the service. This file is usually named pktfilter.log, but again another name could be specified during installation. Logging worked perfectly in Windows 2000, but in Windows XP the log file started at zero bytes on installation and never changed. Again, this was found to be an issue with XP Service Pack 1.

Being a stateless packet filtering program, PktFilter requires pass rules for responses to any queries sent by the desktop. This makes more 'holes' in your firewall for hackers to exploit. If it were statefull, then only packets for an active session would be passed.(4)

Through an NT Logon Script, the service can be installed on the user's computer, the rules file is copied to the appropriate directory, and the service is started. If the user requires any special rules for specific functions, a user text file is appended to the rules file just before the service is started. In addition, special rules for administrator's machines are appended during logon. In this way, the rules file is custom made for each user only if necessary. If many users' log files show the same blocked packets, the correct rule may be permanently written into the default rules file which exists on the login server.

The following text file is effective for Windows 2000 computers using file and printer sharing in a domain environment. Changes can be made as needed by the administrator, and the service restarted. If the syntax is correct, the service starts.

Key to the rules file IP addresses:

192.168.0.50 = This Computer
192.168.0.0/24 = Local Subnet
192.168.2.52 = DHCP Server
172.16.0.2 = Primary DNS Server
172.16.0.4 = Backup DNS Server
192.168.2.100 = Primary Domain Controller
192.168.2.155 = Backup Domain Controller
192.168.2.0/24 = Server Subnet
192.168.0.153 = UNIX Database Server
192.168.0.147 = UNIX Server
172.16.0.50 = UNIX Application Server
192.168.2.132 = Anti-Virus Server
192.168.2.101 = File and Print Server
192.168.2.77 = Exchange Server
xxxx = port specified in a work related webpage

Rules.txt File:

```
#----------------------------------
#  IP Filtering Rules for Desktops
#----------------------------------
#-------------------------------------------------------------
# drop packets composed of small fragments
#-------------------------------------------------------------
```

```
option small_frags on eth0
#---------------------------------------------------------------
# Default Behavior (used if a rule below is NOT matched)
#---------------------------------------------------------------
block in on eth0 all
block out on eth0 all
#---------------------------------------------------------------
# Allow NetBIOS from Local subnet
#---------------------------------------------------------------
pass in on eth0 proto tcp from 192.168.0.0/24 port > 1023 to any port = 135
pass in on eth0 proto tcp from 192.168.0.0/24 port > 1023 to any port = 137
pass in on eth0 proto tcp from 192.168.0.0/24 port > 1023 to any port = 139
pass in on eth0 proto udp from 192.168.0.0/24 port > 1023 to any port 136 >< 139
pass in on eth0 proto udp from 192.168.0.0/24 port 136 >< 139 to any port 136 >< 139
pass out on eth0 proto udp from any port 136 >< 139 to 192.168.0.0/24
pass out on eth0 proto tcp from any port = 135 to 192.168.0.0/24
pass out on eth0 proto tcp from any port = 137 to 192.168.0.0/24
pass out on eth0 proto tcp from any port = 139 to 192.168.0.0/24
#---------------------------------------------------------------
# Allow ALL NetBIOS Responses
#---------------------------------------------------------------
pass in on eth0 proto tcp from any port = 135 to any port > 1023
pass in on eth0 proto tcp from any port = 137 to any port > 1023
pass in on eth0 proto tcp from any port = 139 to any port > 1023
pass in on eth0 proto udp from any port 136 >< 139 to any port > 1023
pass out on eth0 proto udp from any port > 1023 to any port 136 >< 139
pass out on eth0 proto tcp from any port > 1023 to any port = 135
pass out on eth0 proto tcp from any port > 1023 to any port = 137
pass out on eth0 proto tcp from any port > 1023 to any port = 139
#---------------------------------------------------------------
# Allow WINS/DHCP server to respond and query
#---------------------------------------------------------------
pass in on eth0 proto udp from 192.168.2.52 port = 137 to any
pass in on eth0 proto tcp from 192.168.2.52 port = 137 to any
pass out on eth0 proto udp from any to 192.168.2.52 port = 137
pass out on eth0 proto tcp from any to 192.168.2.52 port = 137
#---------------------------------------------------------------
# Allow ALL DNS inbound/outbound
#---------------------------------------------------------------
pass in on eth0 proto udp from 172.16.0.2 port = 53 to any port > 1023
pass in on eth0 proto udp from 172.16.0.4 port = 53 to any port > 1023
pass in on eth0 proto tcp from 172.16.0.2 port = 53 to any port > 1023
pass in on eth0 proto tcp from 172.16.0.4 port = 53 to any port > 1023
pass out on eth0 proto udp from any port > 1023 to 172.16.0.2 port = 53
pass out on eth0 proto udp from any port > 1023 to 172.16.0.4 port = 53
pass out on eth0 proto tcp from any port > 1023 to 172.16.0.2 port = 53
pass out on eth0 proto tcp from any port > 1023 to 172.16.0.4 port = 53
#---------------------------------------------------------------
# Allow the domain controllers to authenticate for me
#---------------------------------------------------------------
pass in on eth0 proto udp from 192.168.2.100 port 136 >< 140 to any
pass in on eth0 proto udp from 192.168.2.155 port 136 >< 140 to any
pass in on eth0 proto tcp from 192.168.2.100 port 136 >< 140 to any port > 1023
pass in on eth0 proto tcp from 192.168.2.155 port 136 >< 140 to any port > 1023
pass out on eth0 proto udp from any port 136 >< 140 to 192.168.2.155 port 136 >< 140
pass out on eth0 proto udp from any port 136 >< 140 to 192.168.2.100 port 136 >< 140
pass out on eth0 proto tcp from any port > 1023 to 192.168.2.155 port 136 >< 140
pass out on eth0 proto tcp from any port > 1023 to 192.168.2.100 port 136 >< 140
#---------------------------------------------------------------
# Allow all HTTPS (SSL) and HTTP inbound/outbound
#---------------------------------------------------------------
pass in on eth0 proto tcp from any port = xxxx to any port > 1023
```

```
pass in on eth0 proto tcp from any port = 443 to any port > 1023
pass in on eth0 proto tcp from any port = 80 to any port > 1023
pass out on eth0 proto tcp from any port > 1023 to any port = 80
pass out on eth0 proto tcp from any port > 1023 to any port = 443
pass out on eth0 proto tcp from any port > 1023 to any port = xxxx
#---------------------------------------------------------------
# Allow inbound/outbound from Mail Server (for Exchange)
#---------------------------------------------------------------
pass in on eth0 proto tcp from 192.168.2.77 port = 135 to any
pass in on eth0 proto tcp from 192.168.2.77 port = 755 to any
pass in on eth0 proto tcp from 192.168.2.77 port = 756 to any
pass in on eth0 proto udp from 192.168.2.77 port > 1023 to any port > 1023
pass out on eth0 proto tcp from any to 192.168.2.77 port = 135
pass out on eth0 proto tcp from any to 192.168.2.77 port = 755
pass out on eth0 proto tcp from any to 192.168.2.77 port = 756
#---------------------------------------------------------------
# Allow Subnet Broadcasts (if you don't, it does a lot of logging)
#---------------------------------------------------------------
pass in on eth0 proto udp from 192.168.0.0/24 port = 138 to 192.168.0.255 port = 138
pass in on eth0 proto udp from 192.168.0.0/24 port = 137 to 192.168.0.255 port = 137
#---------------------------------------------------------------
# Allow inbound/outbound ICMP (Ping)
#---------------------------------------------------------------
pass in on eth0 proto icmp from 192.168.0.0/24 to any
pass in on eth0 proto icmp from any to any icmp-type echorep
pass in on eth0 proto icmp from any to any icmp-type timex
pass out on eth0 proto icmp from any to any
#---------------------------------------------------------------
# Allow inbound/outbound from UNIX Servers
#---------------------------------------------------------------
pass in on eth0 proto any from 192.168.0.153 to any
pass in on eth0 proto any from 192.168.0.147 to any
pass out on eth0 proto any from any to 192.168.0.153
pass out on eth0 proto any from any to 192.168.0.147
#---------------------------------------------------------------
# Allow inbound/outbound from Application Server
#---------------------------------------------------------------
pass in on eth0 proto any from 172.16.0.50 to any
pass out on eth0 proto any from any to 172.16.0.50
#---------------------------------------------------------------
# Allow inbound/outbound from anti-virus server
#---------------------------------------------------------------
pass in on eth0 proto any from 192.168.2.132 to any
pass out on eth0 proto udp from any port > 1023 to 192.168.2.132 port > 1023
pass out on eth0 proto tcp from any port > 1023 to 192.168.2.132 port = 139
#---------------------------------------------------------------
# Allow Messaging Service
#---------------------------------------------------------------
pass in on eth0 proto udp from 192.168.2.101 port > 1023 to any port = 135
pass in on eth0 proto udp from 192.168.2.101 port > 1023 to any port > 1023
#---------------------------------------------------------------
# Allow inbound/outbound from File and Printer Server
#---------------------------------------------------------------
pass in on eth0 proto tcp from 192.168.2.101 to any
pass out on eth0 proto udp from any port > 1023 to 192.168.2.101
pass out on eth0 proto tcp from any port > 1023 to 192.168.2.101 port 136 >< 140
###############################################################################
```

Notes:

- Actual IP addresses are routable. Non-routable given for example only.
- The first three rules are default with the program, and should be left alone.

- After the default rules, only rules that provide exception are processed. Any rule which matches the packet is processed.
- The "any" argument for the IP address in 'pass in...to any' and 'pass out...from any' rules stands for the user's desktop IP address. This allows the same rule file to apply to most machines. The machines have dynamically assigned addresses, and in some cases, may have more than one IP address assigned to the network card.
- Although DNS requests coming through port 53 are usually of the udp protocol, port 53 is reserved for both tcp and udp protocols(5), so the additional rules are possibly necessary.
- ICMP protocol is also blocked inbound at the perimeter firewall. The rules set leaves it open for the local subnet, and allows outbound pings and trace routes.
- Many of the inbound rules could be deleted (closing those ports) if this were a statefull firewall.

The rules file is a growing, changing document. It is approximately 8KB in size.

Risk	Vulnerability
16	network resource identified
16	network resource identified
10	IP name obtained
5	IP address found from name

Figure 4 Screenshot of scan with Pktfilter running using the above rules.txt

This is an excellent result. As a matter of fact when the same scan was performed against an IP address which was not leased, *and had no machine in place to receive the scan*, the results were the same.

Risk	Vulnerability
16	network resource identified
16	network resource identified
10	IP name obtained
5	IP address found from name

Figure 5 Screenshot of scan of no computer at IP address

Obviously in this case, these are false positives.

After Snapshot

The service was deployed to all the users and community desktops using the Windows NT logon script which calls a batch file - updaterules.bat during logon.

Updaterules.bat:

```
@echo off
::---------------------------------------------------------
::Check the Computer Name
::---------------------------------------------------------
IF %COMPUTERNAME% == JSMITH goto install
IF %COMPUTERNAME% == BJONES goto install
IF %COMPUTERNAME% == JDOE goto install
IF %COMPUTERNAME% == CDBURNER goto install
GOTO end
:install
::---------------------------------------------------------
:: Stop the service
::---------------------------------------------------------
net stop "Stateless Packet Filtering"
::---------------------------------------------------------
mkdir C:\Progra~1\PktFlt
copy \\fileserver\loginscripts\packetfilt\exe\pktfltsrv.exe C:\Progra~1\PktFlt\pktfltsrv.exe
copy \\fileserver\loginscripts\packetfilt\exe\pktctl.exe C:\Progra~1\PktFlt\pktctl.exe
copy \\fileserver\loginscripts\packetfilt\exe\pktfilter.log C:\Progra~1\PktFlt\pktfilter.log
C:\Progra~1\PktFlt\pktfltsrv -i "c:\Progra~1\Pktflt\rules.txt" "c:\Progra~1\Pktflt\pktfilter.log"
::---------------------------------------------------------
:: Update the Rules
::---------------------------------------------------------
copy \\fileserver\loginscripts\packetfilt\general.txt c:\Progra~1\Pktflt\rules.txt
if exist \\fileserver\loginscripts\packetfilt\%computername%.txt
 type \\fileserver\loginscripts\packetfilt\%computername%.txt >>c:\Progra~1\Pktflt\rules.txt
::---------------------------------------------------------
:: Start the service...
::---------------------------------------------------------
net start "Stateless Packet Filtering"
::---------------------------------------------------------
:: Done
::---------------------------------------------------------
:end
```

Notes:

- Substitute the actual machine name of the login server for 'fileserver' in the batch file

Once all of the machines in the subnet have the program installed, the administrator must configure the service to start automatically using the services administration tool. This can be done remotely. If this is not done, the service will not be started until the first logon after the bootup. When this is done, the service adds two minutes to the time waiting for the Ctrl-Alt-Del logon window. This is the time that the service is starting. This delay may be addressed when the program is more fully developed.

Once the service is up and running, the log file can be accessed by the administrator through the administrative share. There is a TAIL for Windows program called mTAIL which makes it possible to observe the log file dynamically.(6) This is a handy utility.

Cost is definitely a concern leading one to choose to use PktFilter. As far as bang for the buck, PktFilter is the easy leader. Could a hacker design exploits which take advantage of the openings left in the filter? These are already well known, and well worn paths. This is where a statefull firewall would come into use. If the response pathway were blocked to all but legitimate replies, then we would have a higher degree of protection.

Perhaps the solution would be to combine Pktfilter and a centrally managed firewall system. These are all fairly new products. Future releases of firewalls might have the component problem worked out. For those interested in investigating the options available, Network Computing had a February 20, 2003 article which acted as a buyers guide to centrally managed firewalls.(7) In this article, the author does a good job of explaining the limitations of any single protection strategy. Five firewall systems are compared, sorted and ranked. Licenses for the products average about $50 per end user. This is considerable in a time of reduced budgets. It also is only the beginning, as administrative costs are sure to increase. This is a fact of life today, regardless.

Impact

The systems all scanned with the same four vulnerabilities listed in figure 4. Printers with their own network interfaces are the culprits in our scan reports now. Obviously, printer vulnerabilities are not in the same league as PC vulnerabilities. Six desktop machines were now found to be blocking packets sent to ports or IP addresses they should not have been. Three of these were computers whose configurations were set differently when their users worked in different offices. Two had 'spyware' programs on them, and one was set up to access a POP mail account. The appropriate corrections were made to these machines.

Scans were also performed against the test prototype using nmap for Windows. We had heard that nmap could crash the scanned machine during a scan if configured to do so. While we did not know how to crash a victim computer, every scan we tried against the PktFilter enabled machine left it stable and untouched. No significant processor use was reported by the task manager, and legitimate traffic - including streaming media - was handled by the PC. It should be noted that the scanned machine in all of these tests is a PIII Pentium 700mhz with 256 mb ram.

Another scan was performed using Internet Security Systems' Internet Scanner. The before and after in this case was once again impressive. Even if the scanner *could* ping the scanned system, vulnerabilities listed were low. If it could not ping the system, It did not even include it in the IP addresses to scan list. If the host were then included in the scan list - forcing a scan - the scanner listed the machine as 'not found.'

Finally, we placed the test machine outside the perimeter firewall. System security scan tests are available on the internet at PC Flank(8), SecurityMetrics(9), and Gibson Research(10) among others. Going to these pages gives the user multiple links with which to test the computer he is using. All tests performed against our desktop machine said ports were 'stealthed' and the PC was 'secure.' This would be important for a web server which would reside in a 'DMZ' one step closer to the internet. Of course, the rules file for a web server would close many of the ports left open for our desktop machine.

Every effort should be made to secure any networked computer by keeping up with patches, providing adequate policies, turning off unneeded services, and of course limiting physical access. However, it is only after packet filtering or firewall software is installed that a Windows PC can be said to be truly 'hardened' yet completely accessible to administrative functions.

References:

(1)Sundling, Bob. Why your firewall sucks :). URL: http://tooleaky.zensoft.com/ (April 5, 2003)

(2)How to change the loading order of your Symantec firewall product. URL: http://service1.symantec.com/SUPPORT/ent-security.nsf/docid/2000040412261548 (April 5, 2003)

(3)Zone Labs: Enterprise Solutions. URL: http://www.zonelabs.com/store/content/company/corpsales/intOverview.jsp

(4)Personal Firewall Software. URL: http://wssg.berkeley.edu/public/projects/SecurityInfrastructure/reports/SystemHardening/FireWalls.html (April 5, 2003)

(5)port-numbers. URL: http://www.iana.org/assignments/port-numbers (April 5, 2003)

(6)Phillip, Oliver. mTail - A windows tail program. URL:http://ophilipp.free.fr/soft/mTail.zip (April 5, 2003)

(7)NWC|Review|Security|Defense Starts Here|| February 20, 2003 URL:
http://www.networkcomputing.com/1403/1403f3.html (April 5, 2003)

(8)PC Flank: Make sure you're protected on all sides. URL:
http://www.pcflank.com/
(April 5, 2003)

(9)SecurityMetrics - free Port Scan - Free Server/Firewall Test. URL:
http://www.securitymetrics.com/portscan.adp (April 5, 2003)

(10)Gibson Research Corporation Home Page. URL: http://grc.com/default.htm
(April 5, 2003)

(11)Integrity Overview.pdf URL:
http://www.clearview.co.uk/docs/Integrity_Overview.pdf (April 5, 2003)

(12)Keir, Robin. FireHole. URL: http://keir.net/firehole.html (April 5, 2003)

(13)SourceForge.net: Project Info - PktFilter. URL:
http://sourceforge.net/projects/pktfilter/ (April 5, 2003)

(14)SourceForge.net: pktfilter-users. URL:
http://sourceforge.net/mailarchive/forum.php?forum_id=24642 (April 5,
2003)

(15)eEye Digital Security. URL:
http://www.eeye.com/html/research/tools/nmapnt/nmapNTsp1.zip (April 5,
2003)

Thank You to Bob Templeton, whose initiative, experience and insight proved invaluable.