Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

A 6 - Layer Defense for an IT Professional's Home Network

Penetrating an I.T. professional's home system is even more desirable in the eyes of most hackers. This is due to the fact that I.T. professionals will often have key information on their systems to aid a hacker in penetrating a corporate network. Documents may be present that define a full corporate network architecture. Home systems may have dial-in and VPN information, complete with passwords in plain text. In addition, the I.T. professional may use exactly the same user-IDs and passwords at home that are used at wo...

Copyright SANS Institute

Author Retains Full Rights

AD

A 6-Layer Defense for an I.T. Professional's Home Network

Daniel Crider

11/22/01

I.T. professionals have long understood the need for "hand's on" experience. Most I.T. professionals are now faced with an increasing number of products to support, along with an increasing array of new technologies to understand. In order to deal with this many are now opting to build small home networks or home labs to help them maintain and increase their skills.

Security is becoming a major concern for the I.T. professional, even at home. An I.T. professional is far more likely than a casual PC user to have or to need a high-speed connection to the Internet. For most I.T. professionals this means that their home network is connected via DSL or Cable Modem to the Internet. Unfortunately home systems with high-speed, "always-on" connections are becoming the targets of choice for many hackers. Home systems with high-speed connections are often poorly protected, and still provide excellent "middle" systems to use as launching points for attacks on other systems.

Penetrating an I.T. professional's home system is even more desirable in the eyes of most hackers. This is due to the fact that I.T. professionals will often have key information on their systems to aid a hacker in penetrating a corporate network. Documents may be present that define a full corporate network architecture. Home systems may have dial-in and VPN information, complete with passwords in plain text. In addition, the I.T. professional may use exactly the same user-IDs and passwords at home that are used at work, and these often have elevated privileges (such as Domain Admin, Administrator, or root) on the corporate network. And if the hacker succeeds in penetrating the home system which has a VPN connection in place for the corporate network; then in point of fact the hacker has now found a backdoor into a corporate system.

In late October of 2000, network security of the Microsoft Corporation was successfully penetrated. Although it is not known for certain how hackers gained access, it is believed by many that an employee's home system was the backdoor used by the hackers. [(1) (2)] Computer professionals everywhere should take this as a "wake-up call" that home computer security is a key aspect of corporate network security.

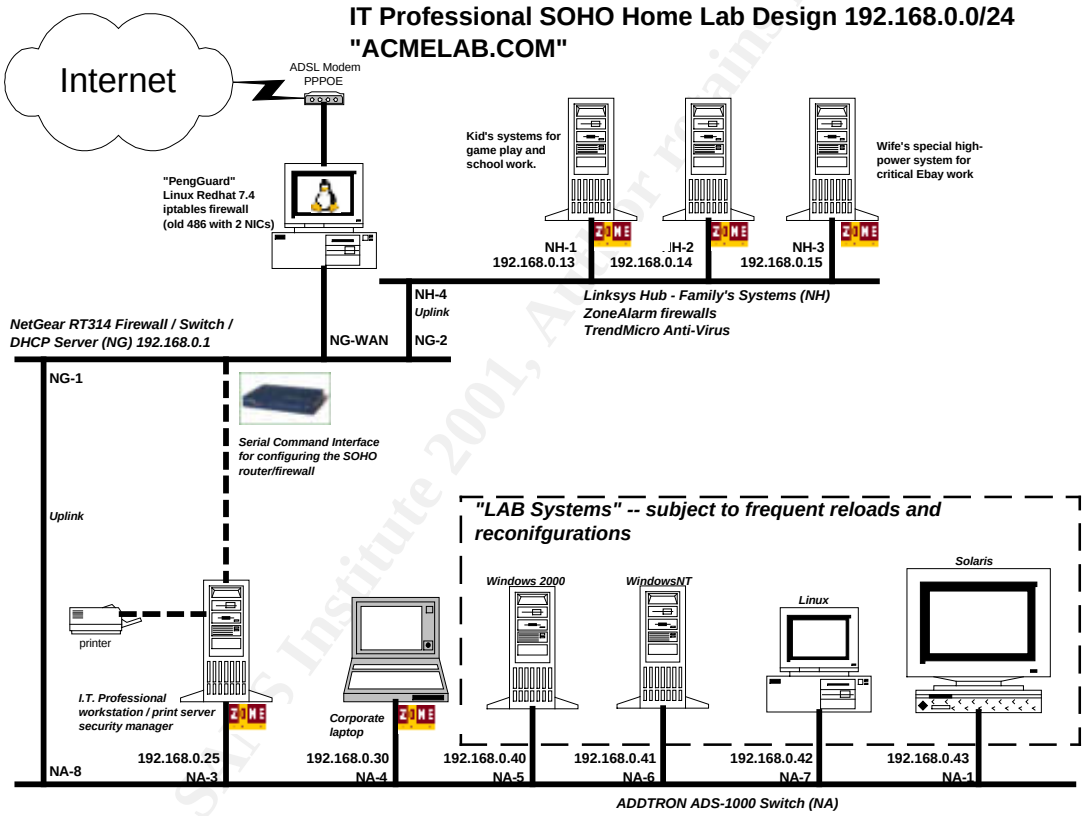
But the I.T. professional has more problems at home than a typical home or SOHO user. His home network will probably contain several computers. Some of these may be for personal or family use. Other systems in his "lab" may require Internet access, yet be difficult to secure due to the dynamic nature of lab machines. These systems will often have the operating system reinstalled frequently. Or major configuration changes may happen on a frequent basis as new products are added for testing or education purposes. Servers, such as telnet, FTP, e-mail, database, and web servers may normally be strongly protected at the corporate office. But in a home lab the same types of servers may be running with no real defenses. Another problem is the fact that more and more I.T. professionals are trying to become proficient in multiple operating systems. An I.T.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

professional's home lab may have Windows9x, WinNT, Win2K, Linux, BSD, Solaris, Apple Macintosh, and/or Novell systems all present.

All of these problems: "Always-on" hi-speed Internet access; personal and professional systems on the same SOHO network; multiple operating systems; and a non-stable environment may make it seem like security will be impossible. But in fact a layered defense can be created that requires relatively little maintenance and provides excellent security. There are 6 recommended layers of defense for an I.T. professional's home network. A 7th optional layer can provide even better defense, but will increase your costs and effort some.

The following network diagram shows a hypothetical SOHO network for an I.T. Professional:



This system has 3 personal or family systems, all running some form of Windows. There is a corporate laptop, and 4 "lab" systems that are expected to be frequently re-configured. There is a single system dedicated to the I.T. professional's regular personal use (documentation, Internet research, banking, game play, etc.) that is also in use as a print server and serves as a security manager for the system. The network architecture consists of 2 Ethernet hubs, 1 SOHO firewall / router, 1 ADSL modem, and 1 Intel PC running as a dedicated Linux firewall router. There are 3 different types of firewalls present. While this configuration may seem complex to some, to many I.T. professionals this is elementary. Many I.T. professionals routinely run between 5 and 10 lab PCs at

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

home, usually on older hardware or on homemade systems. The layered security defense discussed in this paper is designed to be effective, scalable, and inexpensive. It should protect easily between 4 and 20 “lab” machines on a home SOHO network. And if you are running more than 20 lab PCs at home, you are either single, divorced, or your wife is a fellow I.T. professional and the two of you fight over the best hardware. In any case, a home system of that size would need to have upgraded defenses – almost to the same level as a true corporate network. And costs and complexity would increase.

The 6 layers of our I.T. professional SOHO LAN defense include:

- NAT
- a hardware SOHO firewall/router (packet firewall)
- a Linux stateful firewall built on iptables
- personal firewalls (Zonealarm, BlackIce, etc.)
- automated port scanning
- anti-virus software
- vulnerability scanner (optional)

1. Network Address Translation (NAT).

Network Address Translation is a means of allowing computers on a LAN to access the Internet while "masquerading" with the IP address of a host with a suitable address and configuration. It is often used to share one public, routable IP address among hosts located on a LAN behind a masquerading proxy where the local addresses are private and non-routable.

Your internal (SOHO) network utilizes IP addresses reserved by the Internet Assigned Numbers Authority (IANA) for private networks. These TCP/IP addresses cannot be routed on the Internet. These addresses are:

Class Address Range
"A" 10.0.0.0 to 10.255.255.255
"B" 172.16.0.0 to 172.31.255.255
"C" 192.168.0.0 to 192.168.255.255

For example, if your SOHO network was using a network address of 192.168.0.0/24 one of your PCs might have an address of 192.168.0.100. Your gateway to the Internet could be a system with an internal address of 192.168.0.1 and an Internet address of www.xxx.yyy.zzz. The Internet address has probably been assigned by your ISP, and is most likely a dynamic (DHCP) address. If you attempted from your PC to use your browser to surf a web site (www.sans.org is the example I will use here) then your system would try to send a packet to your router (192.168.0.1) on port 80, destined for SANS’s web server (12.33.247.6 is a current address). Your SOHO router could convert this to a “masqueraded” source address of www.xxx.yyy.zzz, port 5000. The SANS web server would send its replies back to your router, destined for port 5000. Your router would know that this port was currently in use by your PC, and would translate it back to 192.168.0.100 port 80.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Using NAT is a great idea for improving your security. Your internal address cannot be used on the Internet, as these IANA privately defined address ranges are non-routable and are programmed to be dropped by all Internet routers. So even if a hacker is able to learn the address of the machine on your local SOHO network, no traffic bound for that address would be allowed on the public Internet.

NAT is an excellent first layer of defense. But it is not enough. A local SOHO network using NAT can still be hacked. But now the hacker must compromise one of your internal systems to use as a bridgehead on your local network. You can find more information on these private TCP/IP addresses by reading RFC 1918 [(3)].

2. A hardware SOHO router/firewall.

There are basically three types of firewalls: packet-filtering, stateful-inspection, and application-level (proxy). For a hardware SOHO router/firewall we are talking mainly about the packet-filtering type. A packet-filtering firewall looks at the IP packets that are incoming and outgoing. Packets can be rejected because they are certain protocols that are being blocked (like an attempted telnet or ftp from the Internet into your SOHO network). Or they can be blocked on other criteria such as source IP address range or the settings of specific flags in the TCP/IP packet (like the SYN flag).

Packet-filtering firewalls are fairly simple and fast. For this reason they lend themselves easily to hardware implementations. There are several inexpensive hardware versions currently targeting the SOHO market. Most of these are now available in the $100-$350 price range. These devices provide not only a firewall, they also provide an efficient means to share a broadband Internet connection (such as DSL or cable modem) to a SOHO network. Most have NAT capabilities built-in. Many also have DHCP and/or DNS capabilities. Many of these also provide a high-speed network switch that allows the SOHO network to be shared to a number of individual PCs – or to hubs. My own favorite device in this category is the Netgear RT314 [(4)] Gateway Router. Similar products are available from Linksys [(5)], Dlink [(6)], and SMC [(7)]. The Netgear RT314 is both a packet firewall and a 4-port 100 mbs switch. It also has the ability to dump CDR (Call Detail Record format) type log records to any Unix system via syslog. And it can establish NAT.

All of these SOHO hardware firewalls have the advantage of being quick to set up, and low maintenance. However, a packet-filtering firewall can be easily fooled. Source addresses can be spoofed, so that a blocked IP range is no longer blocked. And if the firewall is by default allowing packets with the ACK flag to be passed this can be bypassed by an outside hacker.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

3. A “stateful-inspection” firewall.

The stateful-inspection firewall goes beyond packet filtering. It makes sure that traffic is not only formatted correctly (the right IP source and destination addresses, the right flags, etc.), but that it is behaving correctly. For example, traffic on port 80 had better be HTTP traffic, rather than telnet traffic. And even if it “claims” to be HTTP, the traffic needs to act in a manner like HTTP. If a packet claims to be for an existing connection, does that connection really exist? What makes a stateful firewall truly "stateful" is not that it can filter packets, or that it can validate packet data. The key attribute of a stateful firewall is that it maintains a communication history for each connection and can verify when a packet comes in that the connection it claims to be a part of is real.

Linux systems can make relatively inexpensive firewalls. Even better, you can now use them to create stateful-inspection firewalls. Firewall software was built in to Linux kernels from version 1.1 on. It started with code called ipfw, originally supplied by BSD. This was ported to Linux in 1994 by Alan Cox. In mid 1998 the ipchains firewall was developed by Paul “Rusty” Russell and Michael Neuling for the Linux Kernel 2.2. In 1999 Rusty Russell and others developed the “Netfilter” firewall software, which comes with an interface known as “iptables”. This is now included in the Linux 2.4 kernel (including Red Hat 7.2).

You may think that a Linux-based firewall would be a low-performance, low-capability firewall. Not so. Netfilter/iptables is a huge advance over the older ipchains. The new iptables, unlike the ipchains that preceded it, has the capability of being a full-fledged stateful-inspection firewall. Netfilter tracks all of the connections in tables (hence the name of the interface, “iptables”). Iptables tracks and filters connections based on their state. The states are: NEW, ESTABLISHED, RELATED, or INVALID. Even with only a few simple rules, you can achieve a firewall capability previously found only in “top-of-the-line” enterprise firewalls like Cisco, Checkpoint, and Raptor. Other capabilities include:

- Packet-filtering (based on type of packet, source IP, destination IP, flags)
- NAT
- Rate limiting (which helps you prevent Denial of Service attacks)
- Filtering of both inbound and outbound interfaces
- Filtering by the user or process initiating a connection

You may download a copy of iptables from http://netfilter.samba.org/ . The following information on configuring an iptables firewall has come from Rusty Russell’s excellent “Linux 2.4 Packet Filtering HOWTO”. [8] For more information on stateful firewalls see the whitepaper “Anatomy of a Stateful Firewall” by Lisa Senner. [9]

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

To create a simple firewall machine, add the following lines to the file /etc/rc.d/rc.local on a Red Hat 7.2 server:

```
# turn on ipforwarding
echo 1 > /proc/sys/net/piv4/ip_forward
#
# set up NAT
iptables –t nat –A POSTROUTING –o ppp0 –j MASQUERADE
# drop anything from the INPUT table on the ppp0 (internet-side) connection
iptables –A INPUT –i ppp0 –m state --state NEW,INVALID –j DROP
# drop anything coming into the FORWARD table from ppp0 too
iptables –A FORWARD –i ppp0 –m state --state NEW,INVALID –j DROP
```

Note that in the above example it is assumed that the firewall has two Ethernet interfaces:, eth0 and eth1. The eth0 interface is being used to talk to a DSL router with PPPoE. This creates the virtual interface ppp0. The /etc/rc.d/rc.local script file is the place to add anything to the startup commands for a Linux system. These commands will be run after all other startup scripts run. The first line, "echo 1…" turns on IP forwarding. The first iptables line, "iptables –t nat…" sets up Network Address Translation. The second iptables line, "iptables –A INPUT…" creates a stateful filter on the INPUT table. Since the states that will be dropped are "NEW" and "INVALID" these means any attempted connections from the Internet side (ppp0) will be dropped. Connections started from your SOHO network side (eth1) will be allowed. The last iptables line does the same thing, but deals with the FORWARD table. For more information on the various tables in the iptables firewall, see the previously referenced article: "Linux 2.4, Packet Filtering HOWTO" by Rusty Russell. [8]

These commands will work well with RedHat 7.2. They should also work on any other Linux 2.4 kernel system. The iptables firewall software can be installed on earlier kernels (see the "HOWTO" article referenced above), but why bother? If you are taking an old 486 or old Pentium to dedicate for a firewall, just download RedHat 7.2 and install it. Since this will be a dedicated firewall a custom installation will probably work best. You don't even need Xwindows (much less KDE or GNOME) if you make the required modifications to the /etc/rc.d/rc.local file listed above (use the vi editor) and set up the networking properly with the /etc/sysconfig/network file. In fact, if you are careful about setting up a minimalist system, you should be able to get it installed on 700 meg disk drive or smaller. So a 486, 2 ethernet cards (10 mbps is fine for DSL or cable modem), a small disk drive, and an old monitor – and now you have a stateful firewall.

The double firewall approach makes the hacker's job much harder. Now he must compromise the outer Linux firewall, and the inner hardware SOHO firewall. To make this more secure I would disable all services other than iptables on the Linux firewall. Make sure telnet, FTP, Xwindows, SSH, etc. are all shut down. This will require you to keep an old monitor on the system and to manage it locally, but it greatly simplifies your iptables configuration. On the other hand, it is possible to configure iptables to allow

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

something like SSH to remotely login from your local security manager station. See the previously referenced "HOWTO" for details and examples. [8]

4. Personal firewalls on each system.

A "personal" firewall is a small program that is running on an individual system, basically monitoring and restricting attempted connections to this system from other systems. If the system you are on is a client and not a server, there is normally no need for other systems to connect to this system from other systems. Exceptions to this would include client systems that would be remotely managed, and client systems that might be being used for some type of server function (such as game play for a home PC).

You may wonder why yet another firewall would be needed. After all, we are already using two, and that plus NAT should be able to block any outside attack. True, but there are other ways to compromise an internal system. The most common would be either a Trojan-horse program accidentally downloaded (or received in e-mail), or a piece of hostile Java code run on a compromised web server. Any of these could create hidden backdoors and servers on internal systems, and then could attempt to perform a scan of the internal network. Personal firewalls will not only block a compromised system from attacking your other internal systems; they will also help alert you to the fact that one of your internal systems has been compromised.

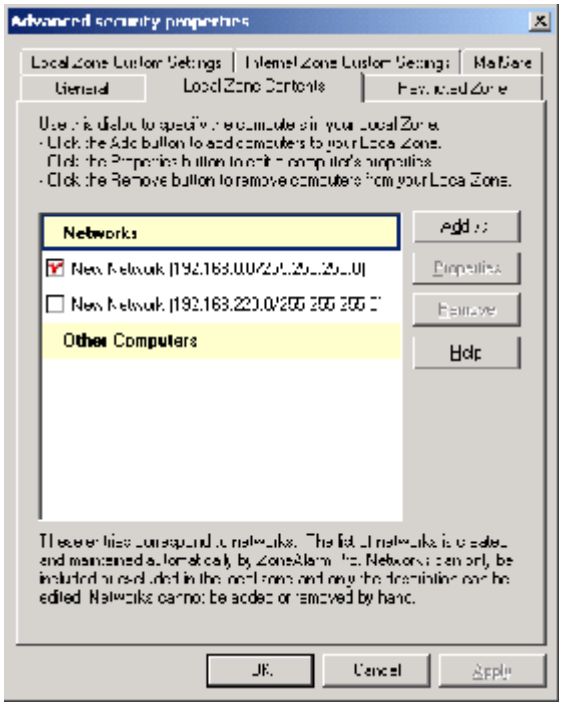
Windows Systems. In the Microsoft Windows world (Windows95, Windows98, WindowsME, WindowsNT, Windows2000, WindowsXP) there are several cheap or very low cost (less than $50) personal firewalls. These programs are usually designed for the novice user, and are very easy to install and to configure. Products in this category include ZoneAlarm (www.zonealarm.com), Tiny Personal Firewall (www.tinysoftware.com), and Black Ice (http://www.networkice.com/).

I myself run ZoneAlarm Pro (approx. $40 for a single-user license). ZoneAlarm also has a less-capable free version. The program works by letting the user define the local trusted (network) zone. Then users are allowed to configure which programs have access to either the local zone or the Internet. Programs are viewed as both clients and potential servers.

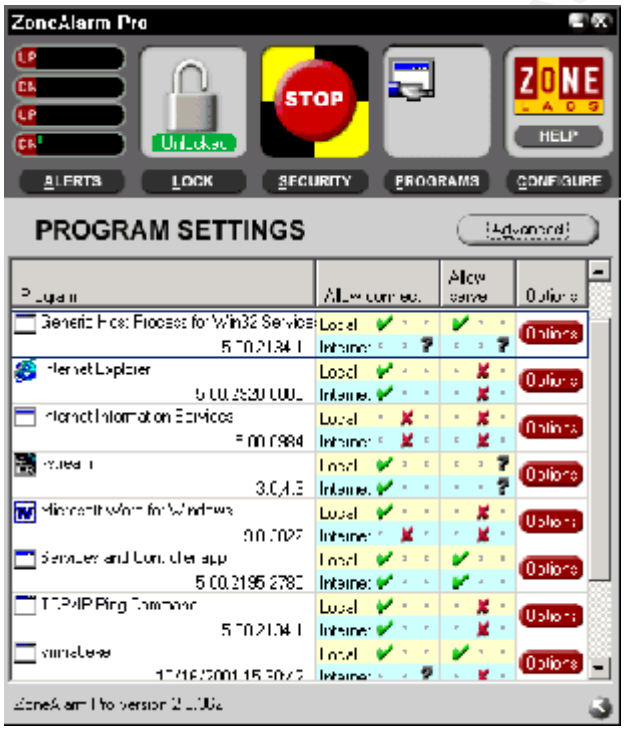
ZoneAlarm is especially effective on Windows systems, in that it will not only warn you if your Windows PC is under outside attack, it will also warn you if a program on your Windows PC is attempting an outside communication – a communication that may not have been authorized or intended by you. This can be your first indication that one of your Windows systems has been compromised. And ZoneAlarm will halt the breach instantly if it has been correctly configured.

The first step is to define the local "trusted" network. This will be your internal SOHO network. If you are using NAT, it should be one of the IANA private addresses. In the example shown here, the SOHO network is 192.168.0.0/24.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.



Once you have defined your trusted local zone, you need to define which programs are authorized to communicate with other systems. Each program will need to be authorized



as either a server or a client (or both); and authorized as to local and/or Internet communications. An “X” means the program is not allowed to connect. A green checkmark means that it is allowed. A “?” means the next time the program is attempted

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

to be used in that manner (client or server) in that zone (local or internet) the user will be asked if this is alright. Don't worry about forgetting a program. The first time a program tries to communicate you will prompted with a question about permission. You can go back and revise your settings at any time should you get one wrong. You will know that you got a program set wrong, if something no longer works. Always err on the side of caution. Deny Internet permission first as a general rule, especially as a server.

Linux Systems. For Linux systems, the very best "personal firewall" is the free Netfilter/iptables firewall already mentioned. This can be downloaded free from http://netfilter.samba.org/, and has already been discussed in the stateful firewall section of this document. Now though our aims are different. We are no longer trying to take a system and make a bridge between the SOHO netework and the outside world. Instead, this is a client system, which we wish to block all external connections to.

The following information on iptables was found in an excellent article by Josuha Drake in LinuxWorld. [10]

To be completely safe, one should block all incoming traffic. The following line will accomplish this:

```
iptables -A INPUT -p tcp --syn -j DROP
```

This will allow the user of the computer to perform all your normal Internet activities. You will be able to browse the Web, ssh out, or chat with a colleague on ICQ. Anything from outside trying to connect to your Linux box via TCP/IP, will simply be ignored. This is the best approach.

If you desire to allow remote management (remote login like telnet, ability to copy files like ftp, etc.) it is recommended that you use SSH. SSH by default uses port 22. The following commands to iptables will allow this:

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

```
iptables -A INPUT -p tcp --syn --destination-port 22 -j ACCEPT
iptables -A INPUT -p tcp --syn -j DROP
```

Now that you have opened up your Linux system to SSH, it is a good idea to restrict connections, so that only a few stations on your SOHO network have this capability. The following command would allow a system at the address of 192.168.0.100 to connect to you via SSH on port 22:

```
/sbin/iptables -A INPUT -p tcp --syn -s 192.168.0.100/24 --destination-port 22 -j
ACCEPT
/sbin/iptables -A INPUT -p tcp --syn -j DROP
```

When you create an iptables-based firewall, each chain (for simplicity's sake, each line) will be read sequentially. Thus, it is possible to have the previous configuration of only one machine having rights to connect via SSH, and to run a public Web server. This could be done with the following commands:

```
/sbin/iptables -A INPUT -p tcp --syn -s 192.168.0.100/24 --destination-port 22 -j
ACCEPT
/sbin/iptables -A INPUT -p tcp --syn --destination-port 80 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --syn -j DROP
```

Solaris, HPUX, and BSD systems. Solaris systems (both Solaris X86 and Sparc) and BSD systems cannot run iptables. But there is a free personal firewall for them as well. This is known as ipfilter. It is not as easy as iptables to set up, but it is not too bad. This can be downloaded from http://coombs.anu.edu.au/~avalon/ip-filter.html. The ipfilter firewall has the very similar capabilities to iptables, include the ability set up NAT and to filter out non-established TCP packets (i.e. stateful filtering). Good documentation on how to use ipfilter can be found at several web sites. The http://www.charvolant.org/~doug/network/html/node15.html website discusses using ipfilter on Solaris systems. Another excellent site is http://www.obfuscation.org/ipf/ipf-howto.txt. IP-Filter, if installed as a package, puts its binaries and man pages under /opt/ipf and the configuration files under /etc/opt/ipf. Like most firewalls, ipfilter is "rules-based". However, reading the rules and understanding them is slightly different from iptables. The rules are processed from top to bottom, each one appended after another. This quite simply means that if your total /etc/opt/ipf/ipf.conf is:

```
block in all
pass  in all
```

The ipfilter firewall does not stop at the fist rule that matches. It reads all the way thru the ipf.conf file. ***And then it applies the last rule that matched.*** In this case it would apply the second rule:

```
pass  in all
```

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

which would pass all traffic. So when you write your ipfilter rules in the ipf.conf file, make sure you bear in mind that the last rule that matches the packet is the one that will take effect. With that in mind a simple ipf.conf file that would block all attempted incoming connections would be:

```
# TCP/UDP Protocols
#       Allow TCP/UDP requests to go out and keep the results
#       flowing back in.
pass out log on le0 proto tcp/udp from any to any keep state
#
# block all TCP packets with only the SYN flag set (this is the first
# packet sent to establish a connection).
#
block in proto tcp from any to any flags S/SA
#
```

Bear in mind that "le0" is the default Ethernet device name for a Sun Sparcstation running Solaris. SolarisX86 (Intel version) will use different device names depending on the type of Ethernet card. The default for a 3com 3C509 card is "elx0". You can confirm your Ethernet device name with the command:

```
ifconfig –a
```

Apple Macintosh Systems. I do not have any personal experience with these systems or products, but several products are available. These include IPNetSentry (http://www.sustworks.com/site/prod_ipns_overview.html) and Norton Personal Firewall for Macintosh (http://www.symantec.com/sabu/nis/npf_mac/). Your objective should be the same as with a Windows or Linux client system. Allow out all communications. Be specific and very restrictive on what is passed in.

Novell Netware Systems. Netware systems are designed to be servers, rather than "personal" systems. The legacy Novell systems were extremely efficient file and print servers – that ran on the IPX/SPX protocol instead of TCP/IP. This made them relatively "hack-proof" from the Internet (but still vulnerable to internal attack). After Netware 4.x it became easier to configure TCP/IP on Netware systems though, and by the time Netware 5.x came about it was relatively simple to configure a "pure TCP/IP" Netware server. Novell now has Webservers (FastTrack), DNS servers, DHCP servers, Proxy servers and their own type of firewall (BorderManager). These systems are feature-rich, and can be just as vulnerable as any other Unix or Windows TCP/IP system to Internet-based attacks. Unfortunately there does not seem to be any low-cost or public domain software designed as a TCP/IP filter or personal firewall for Netware Servers. BorderManager from Novell is designed to be a full-service Internet firewall for an Enterprise, and is priced as such. If you are running a Netware TCP/IP server on your SOHO network you will have to depend on your other defense layers (packet-filter firewall, stateful-inspection firewall, and port scanning (discussed later in this paper)) to defend these systems. Combine these with good standard security practices (strong passwords, regular audits, etc.) to defend any personal Netware servers you may be

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

running. For more information on the new TCP/IP capabilities of Netware 5 see the 1400+ page reference: CNE Netware 5 Study Guide from Osborne/McGraw Hill. [11]

Personal Firewall Summary. Any system on your SOHO network that is a permanent system (i.e., not one of the "lab" systems that are frequently reconfigured and reloaded) must have some type of personal firewall software permanently installed and configured on it. Something which blocks either all attempted incoming connections, or something that at least restricts such connections to a specific type and to a specific computer or group of computers on your local SOHO network. Even in the case of the "lab" systems, once you practice setting these up a few times and develop some default configuration files they can be secured fairly quickly. If some of your lab systems are going to be local test servers (e-mail, web, ftp, etc.), then set up a simple rule-set that restricts connections only to your local SOHO network.

For more information on personal firewalls in general see the "Home PC Firewall Guide" by Henry Stephen Markus [12] or the whitepaper "Personal Firewalls/Intrusion Detection Systems" by Sean Boran. [13].

5. Automated Port Scanning.

How do you know what your systems are doing as far as their network communications go? Who are they communicating with? Have any back doors or Trojans been installed? Do you have a server service such as telnet or FTP turned on without realizing it? Is your PC a webserver? (IIS is installed ***by default*** in most Microsoft Windows systems!).One of the best and easiest indicators is to find out what TCP and UDP ports are open on your systems.

This is known as doing a "port scan". It is the same method used by hackers to find open doors on your systems that they can exploit. Think of it as checking to make sure all of the doors are locked on your house from the outside.

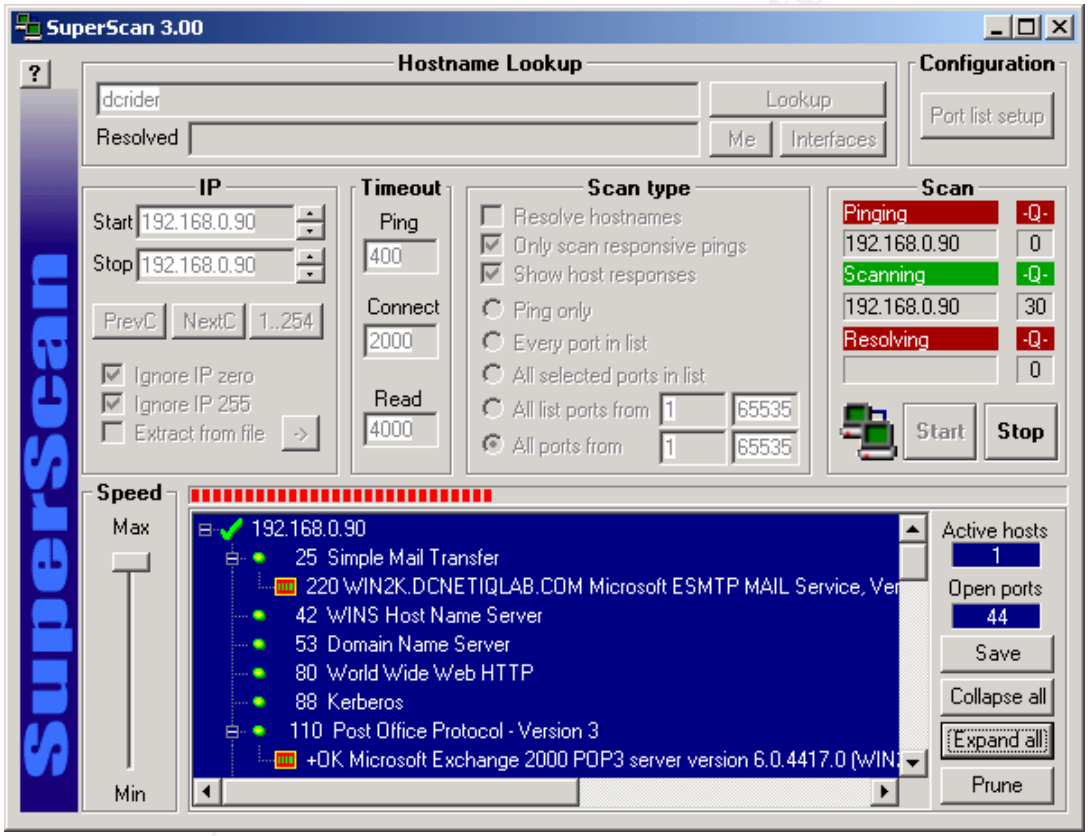
The first step is to perform a baseline scan. This is best done right after an operating system is installed. When you run this against a system that you have been using for a while, you may be very surprised at how many ports are open. When you run against a freshly installed system you find out what ports are open by default. For example, most Windows systems have ports 135 and 139 (NetBIOS) open by default. Windows 2000 systems always have 445 open. Other common ports are:

- 21 for FTP
- 22 for SSH,
- 23 for Telnet
- 25 for e-mail
- 53 for DNS
- 80 for web servers
- 110 for pop3 e-mail server

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

The best way to perform the scan is using a command-line tool. That way you can automate scanning with a script, and output the results to a text file. You can later run a comparison test against the output from the baseline. In Unix the command to compare two files is “diff”. On Windows systems the command “comp” will compare two files. If the output is different have the script send you an e-mail, or place some kind of warning in a log file.

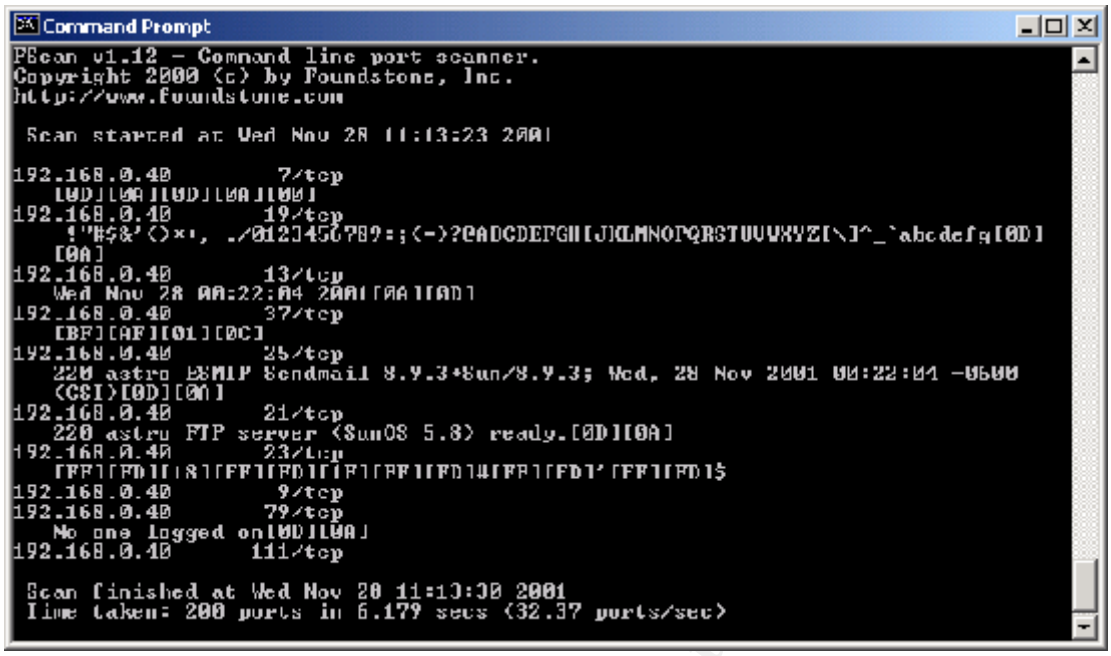
One of the easiest port scanners to use for Windows is the program SuperScan. If you are not familiar with port scanning you may want to start with this one. It can be downloaded for free from www.foundstone.com. It has the ability to output a text file with the scan results, but is not designed to run from the command line. This is a GUI tool. This makes it very useful as an educational tool, but does not lend itself to automated scans with scripts.



Better tools for command line scanning are nmap and fscan. Nmap is especially good, and may be downloaded for free from http://www.insecure.org/nmap/. It even has a new graphical interface for X-Windows (nmapFE). However it is available at the moment only for Linux systems. If fact, it comes free with RedHat 7.2.

Fscan is available from www.foundstone.com for free. This system will run on Windows workstations and servers. A sample output of one of its port scans is shown below:

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.



Set up your scripts to run your scans from a standard system on your SOHO network. This workstation needs to become your security manager for your SOHO network. You will need to make changes on all of your personal firewall settings on your systems to allow traffic from this station to be received.

Once you find open ports on your systems, find out why they are open. The fport tool from www.foundstone.com can help identify what process or service on a Windows system has a port open. The lsof utility, available from http://www.ja.net/CERT/Software/lsof/ can do the same thing on a Unix system. On Unix systems you will also need to check the /etc/services and /etc/inetd.conf files.

A Warning About Port Scanning. Port scanning is an excellent, easy, and fast way to see what has been opened on some of your systems. It is also a standard technique used by hackers to search for vulnerabilities in target systems. For that reason, many ISPs have very strict restrictions on port scanning. It is often considered a hostile activity, and can cause an ISP to revoke your access rights. Always make sure you only scan your own network. Never scan anyone's system without their written permission, and never attempt to scan another system via an ISP. Also be very careful about using such a technique on a corporate network. Some scanners can overload network segments with traffic, depending on their settings. Bear in mind that if you have a VPN connection to your corporate network, you are for all practical purposes on the corporate LAN. If both you and your company are using the same private IP address scheme, you could easily run a scan on the corporate system by accident.

6. Anti-Virus Software.

Although the anti-virus software has long been a security subject, some may not think of it as being related to crafting a defense against hackers. They should. Most high-quality AV software not only detects viruses, it also looks for Trojans and worms. The first

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

indication that a system has been infected by a Trojan might come from an AV package. This will be especially important for any Windows systems on your SOHO network that are used to frequently browse the Internet or to receive e-mail.

Some of the vendors that provide excellent anti-virus software include Norton (Symantec, www.symantec.com), Trend Micro (www.antivirus.com), and McAfee (Network Associates, www.mcafee.com). Whatever AV software you choose make sure you get frequent updates to the signature files. Updating once a week is about the minimum frequency. You may not want to bother with AV software on your "lab" machines. This is fine, so long as you do not access the Internet with them or receive e-mail with them. But if either of those is true, even the lab machines must be protected. This is more true with Windows systems than with Unix, Linux, Macs or Netware systems. Viruses exist for these other platforms too, but they are much less common. Be aware that even a non-Windows system such as an Apple Macintosh can be infected with the same macro viruses that attack Windows systems though if you are using a macro-enabled application such as Microsoft Office X for the Mac. And hostile Java from a compromised web site can run on most browsers – including the Netscape, Mozilla, and Konquest browsers that are often used on Unix and Linux systems.

7. Vulnerability Scanners (Optional).

Hackers find new vulnerabilities into systems every day. Sometimes several major hacks are published each week. And vendors like Microsoft, RedHat, Sun, and Novell scramble to keep up. Vendors are usually pretty good about announcing a vulnerability as soon as it is discovered, and releasing patches to block up the holes as fast as possible. The problem for I.T. professionals is keeping up with all of these vulnerabilities and fixes. Fortunately there are automated systems to help.

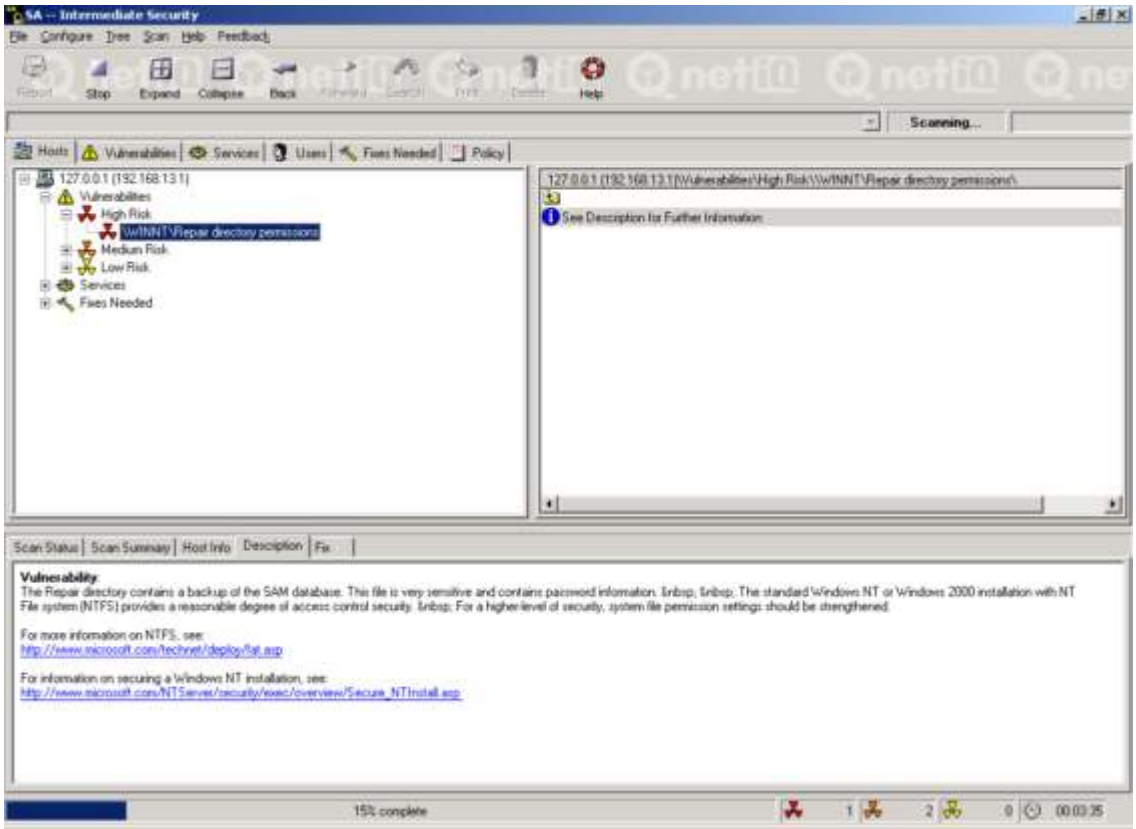
One of the best commercial systems for this is NetIQ's Security Analyzer. This program is designed to run from a Windows platform, but can scan Windows9x, WindowsNT, Windows2000, Linux, and Solaris systems. It installs easily, and auto-updates the known vulnerabilities database on a scheduled basis. Each detected vulnerability has a report that explains how to fix the problem, and where to go for any required patches. The product works best when you set up a server to run SA (your security management workstation) and install an agent on the target systems. However, SA can still run scans without agents. They are just not as detailed and cannot check certain items, like registry settings on a Windows system. Another major capability of Security Analyzer is the ability to run a comparison scan against a previous scan. This allows you to quickly spot something that may have changed on your system. You can customize the scans to determine which tests should be run, or which groups of tests. Scans can also be scheduled to happen on a regular basis.

Another unique factor in Security Analyzer is the ability to program your own tests. These can be written in either Perl or C++. Thus if there is a specific new vulnerability that you have just learned about and you would like to write your own scan for before NetIQ comes out with one, you can quickly set up your own. For those I.T. Professionals who have done any scripting or visual basic work, writing a test file (called "bullet files")

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

for Security Analyzer is relatively simple. The product comes with a nice tutorial on this, along with several examples.

The following is a screen shot of Security Analyzer while it is running a scan.



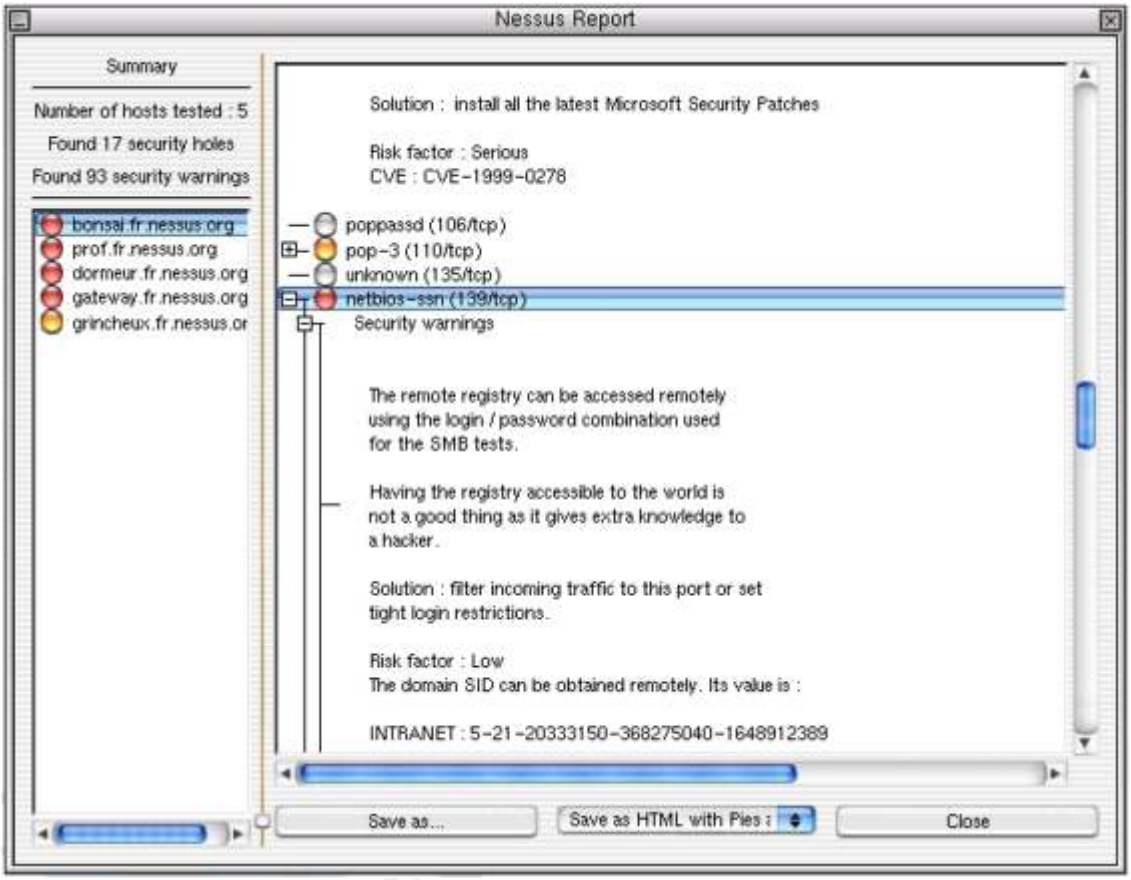
Nessus is a public domain security scanner that can be downloaded for free from www.nessus.org. The Nessus project started in 1998. It is a pure client / server system, i.e. you must have a nessus server and run the client software on the systems you plan to attack. Although Windows client software is available, you must run the server on a POSIX compliant system (GNU/Linux, Solaris, FreeBSD, etc.). Nessus outputs its reports in HTML, ASCII, Postscript, or a format that the Nessus client can read ("NSR"). Nessus runs its scans via plug-ins, with major security tests grouped together in a plugin "family". Like Security Analyzer, Nessus has the ability for you to write your own security test ("attack") in a special scripting language. Attacks can also be written in C. Nessus does not believe that the target hosts will respect the IANA assigned port numbers. This means that it will recognize a FTP server running on a non-standard port (31337 say), or a web server running on port 8080.

Although the Nessus project is a not-for-profit organization they have worked hard on their vulnerability scanner and strive to keep it up to date. It has received excellent reviews in tests against commercial systems. You can actually use a Nessus system over the Internet to check one of your systems. Go to the

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

https://secure1.securityspace.com/smysecure/index.html web site and run a check on one of your systems. That site is powered by Nessus.

A screen shot of a Nessus report.



Putting it all together.

Use the following steps to assemble your defense:

1. Build your Linux Netfilter/iptables firewall.
2. Configure your SOHO pack firewall
3. Make sure NAT is working on either the Linux or SOHO firewall (or both).
4. Set up one of your "permanent" systems as your security manager. Run a port scan on all of your permanent systems. Close any open ports that you can by shutting down unneeded services. Run the scan again and save as a baseline. Set up some scripts for automated scanning.
5. Set up your personal firewalls on all "permanent" systems. Setting up a basic personal firewall is a good idea for the lab systems too. Make sure your designated security manager system is authorized to do port scans and remote logins.
6. Deploy anti-virus software on all of your "permanent" systems.
7. Set up your Security Vulnerability scanner. Use your security management system to scan from. Make sure you have adjusted your settings on all

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

"permanent" systems personal firewalls to allow scans. Install client agents where required.

Testing your firewall defenses.
There are several Internet sites that will run a free automated attack on your systems. Some of the most well known are http://hackerwhacker.com/, www.grc.com (use the ShieldsUp link), and https://secure1.securityspace.com/smysecure/index.html. Access these links from all of your "permanent" systems, and from any of the "lab" systems on your SOHO network that you wish to check. You should know in short order whether or not your 6 layers of protection are in place properly.

Final Comments.
This design for a 6-layer defense should serve I.T. professionals well for SOHO networks. It looks complex, but actually should take less than a day to set up. Costs should be low, and maintenance effort should be minimal. The amount of protection placed on the "lab" systems (anti-virus software, personal firewall software, port scan baselines) is up to the user. Applying all of these is best, but so long as the "permanent" systems have all of these protections on them and so long as the "lab" systems are not used to access the Internet and have a short life cycle (frequently re-installed with new operating systems) I. T. professionals can get away with leaving them relatively exposed. However, setting up personal firewall software on these "lab" servers should be seen as an absolute minimum.

I.T. professionals should also not be under the impression that the steps outlined here, in and of themselves, are sufficient to protect a corporate network. They are not. Some of the subjects that would need to be addressed in depth for corporate protection include the following:

- Corporate Security Policy
- Password strength testing
- Regular Auditing of log files
- Regular Auditing of ACLs (file, directory, and registry permissions on Access Control Lists)
- Regular Auditing of Accounts and groups
- Automated Security Management Systems to centralize alerts and take responses
- Intrusion Detection Systems
- Dial-in and VPN protection
- Database Security
- Encryption of sensitive files, directories, and network traffic
- Incident Handling Procedures

The I.T. professional is a critical link in the chain of computer security, and should realize that protecting his home network is critical for both him and his company. This 6-layer defense should help to accomplish that.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

References:

(1) ZDnet News: "Microsoft Hacked! Code Stolen?" by Ted Bridis and Rebecca Buckman, October 27, 2000. URL: www.zdnet.com/zdnn/stories/news/0,4586,2645850,00.html (11/27/01)

(2) CNN.COM: "Analysis: Home workers can imperil systems", by Jaikumar Vijayan And Carol Sliwa, November 7, 2000. URL: www.cnn.com/2000/TECH/computing/11/07/home.workers.idg/ (11/27/01)

(3) Internet Network Working Group RFC 1918, February 1996. URL: www.cis.ohio-state.edu/cgi-bin/rfc/rfc1918.html (11/27/01)

(4) Netgear Products. URL: www.netgear.com/product_view.asp?xrp=4&yrp=12&zrp=55 (11/27/01)

(5) Linksys Products. URL: www.linksys.com/products/product.asp?prid=20&grid=5 (11/27/01)

(6) Dlink Products: URL: www.dlink.com/products/broadband/di704/ (11/27/01)

(7) SMC Products: URL: www.smc.com/index.cfm?action=products_choose_product&cat_id=4&prodCat=Broadband%20Routers (11/27/01)

(8) "Linux 2.4 Packet Filtering HOWTO", by Rusty Russell, August 2001. URL:http://netfilter.samba.org/unreliable-guides/packet-filtering-HOWTO/ (11/27/01)

(9) White Paper: "Anatomy of a Stateful Firewall" by Lisa Senner. May 9, 2001. URL:http://www.sans.org/infosecFAQ/firewall/anatomy.htm (11/27/2001)

(10) LinuxWorld, "10 minutes to an iptables-based Linux firewall", by Jousha Drake, 2001. URL: www.linuxworld.com/site-stories/2001/0920.ipchains.html (11/28/01)

(11) CNE NetWare 5 Study Guide, Berkeley, Ca., Osborne/McGraw-Hill, 1999. pages 551-588, 673-682.

(12) "Personal Firewall Guide" by Henry Stephen Markus, November 25, 2001. URL: www.firewallguide.com/ (11/28/01)

(13) White Paper: "Personal Firewalls/Intrusion Detection Systems", by Sean Boran, September 28, 2001. URL: www.boran.com/security/sp/pf/pf_main20001023.html (11/28/01)

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 23rd, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced