Global Information Assurance Certification Paper

John Lutheran
jcluthe001

GSEC Practical Assignment
Version 1.2e

Original Submission

Sandia National Laboratories

My Home Setup

Since attending the SANS 5 day training course on security I have made several changes to my home computer configuration. First, I will map out my home network and how it was vulnerable; next I will list the changes I have made after taking the course.

We will start with the equipment list:

1. My personal computer. Pentium III, Windows 2000 & some decent hardware.
2. RedHat Linux PC. Pentium II RedHat 7.1
3. Laptop. Dell Latitude Pentium II RedHat 7.1
4. A cheap 100base-T HUB

The old configuration was simple. I connected my DSL modem to my PC and shared that connection via Windows 2000 Internet Connection Sharing. Microsoft describes how this is done at the following link: http://www.microsoft.com/windows2000/en/professional/help/conn_ics.htm . I found this to be the easiest way to get Internet access from all of my PC's. After my ISP hooked up my DSL connection (which was painful), I rushed to get everything set up without thinking about security. The result of my haste was an insecure configuration. I was using the default or 'out of the box' configuration of Windows 2000. My passwords were not very strong. I had not specified any security options on any level, and I was leaving my PC on all of the time. I was basically a sitting duck on the Internet. One morning, I woke up to find all of the accounts on my Windows 2000 PC locked out. Someone had been trying to get access to my PC via the Internet. I called my ISP and to my surprise, they were not very concerned about this attempted break-in. I expected some immediate advice on how to prevent Internet attacks and what I got was. 'You should get a firewall.' No brand names, no advice on which O/S is the safest for Internet use, nothing. I was totally on my own. (This ISP has since gone Bankrupt…I wonder why?) This sad

attempt at tech support prompted me to start shutting down my PC when I was not using it. I also began reading about Internet security and 'hacking.' I did find many different hints and configuration guides on securing Windows 2000 and I did make some changes including using stronger passwords and upgrading my Windows configuration with many of Microsoft's security updates. I used Microsoft's Windows Update website to acquire these updates. This was the best way I could find to get the correct updates for my particular PC. Access to Windows Update is free and simple to use. The URL is: http://windowsupdate.microsoft.com/ .

After attending the SANS Security Essentials course, I decided to re-think my configuration. I had many choices to make; this is how it all turned out:

First, I decided to use RedHat Linux to share my DSL connection with the rest of my home network. Linux is and always has been available for free. This appealed to me because this is my home configuration and I wanted to keep costs down. My interest in Linux also contributed to my decision. The RedHat install went pretty easy. I simply read some Readme's and hardware compatibility lists, then booted from the CD and answered many questions about my hardware and what kind of networking I wanted to use. I started with just one network card in the Linux PC and only TCP/IP for the network protocol. Linux wants to use this protocol by default and I had no need for any other types of protocols. I was trying to keep the install as basic as I could. This was on advice from many newsgroups and message boards. I found most of my help on all of the subjects in this paper on Google. (http://www.google.com)
I configured my network settings according to my ISP's instructions. At this point, only this one RedHat Linux PC had access to the Internet. I waited to install the second network card in my Linux PC. I had also read that it is best to secure the Linux PC itself before trying to make it into a firewall. So, I made sure there were no extra network services running. I do not run a website or a public ftp site, so I made sure these services were not running. The only service I left running was the SSH. SSH or 'secure shell' is an encrypted telnet client and server which allowed me to connect to my home network for work. SSH also includes SFTP which is an encrypted file transfer client and server. I use SFTP to send files home from work.

The next order of business was installing and configuring a firewall. In the SANS course, many different types of firewalls were discussed, I decided to start with RedHat's support page for help configuring my own firewall. RedHat did have some information on how to configure a firewall on their O/S, here is the link to the page I read: http://www.redhat.com/support/docs/tips/firewall/firewallservice.html.

While this script did work, it was a little difficult to configure. I am not an expert when it comes to Linux and I needed something a little more 'user friendly.' I decided to look

further. The next few days of searching the web using Google (http://www.google.com), I found a web site that proved to be very helpful. The site, http://www.linuxfirewall.org/ is a great resource for many different types of Linux firewall tools. There were many different free resources and I read about each one. Of all the tools listed, I liked the 'Seattle Firewall' best. This tool can be found at: http://seawall.sourceforge.net/.

The Seawall tool can be installed via RPM. I downloaded the package: seawall-4.1.-1.noarch.rpm and installed it as per the instructions in the documentation. The documentation can be downloaded for the same web site and is essential to the successful use of this tool. The requirements are:

1. A computer running Linux with the 2.2x kernel. This tool is architecture independent meaning you can run the tool on just about any type of hardware.
2. IPCHAINS. This package comes with most distributions of Linux. RedHat 7.0 has IPCHAINS included.
3. For Internet access sharing, two connections are needed. An outside connection (analog modem, cable or DSL connection), and an inside connection (a network card connected to the other computers that will use the Internet.

To install the RPM, (at a command prompt with root privileges) type:

rpm –ivh seawall-4.1.-1.noarch.rpm

The install creates a file in the /etc directory named 'seawall.conf' (see appendix A) and a directory in /etc named 'seawall' (/etc/seawall/). The file /etc/seawall.conf is where I configured all of my firewall settings, (see the config file appendix a).

The seawall tool is extremely versatile. There were many options I did not need. What I liked best was the fact that I can configure everything in one file. I just filled in the blanks I needed and left the rest alone. I chose to allow incoming SSH connections from the Internet so I could connect to my home network from work. Other than SSH and SFTP, I blocked everything I possibly could. I did not need any file services like FTP or NFS. I also configured my Linux PC so the default web server did not run at startup. Web servers are targets for DOS or 'Denial of Service' attacks. Basically attackers will send a request to the web server, which confuses the server into giving up information or just giving up in general. Next the attacker has root access to the machine and things go really bad from there. My ISP does not support my running a web server for any kind of commercial gain and I don't have much interest in running one anyway. Once I edited the Seawall configuration file, I was ready to use my Linux PC to share the DSL modem with the rest of my network. I installed a second network card and configured the IP address to connect my entire home network via a small hub. Under the Seawall configuration, I was expected to have one 'Live IP' provided by my ISP and as many 'Private IP's' as I wanted. I used 192.168.0.x s my subnet. Most of what I read on Google suggested this

type of configuration. It is simple and it works. I used 192.168.0.1 (IP) 255.255.255.0 (subnet mask) for my Linux PC. For my other two PC's I used 192.168.0.2 and 192.168.0.3 with the same subnet mask. On the two PC's that reside 'inside' the firewall, I had to add a 'Default Gateway.' This was, of course, the IP address of my Linux PC (192.168.0.1). A gateway is just that, a gateway to the Internet, a computer or other device that shares the connection to the outside world. With much doubt, I then launched Internet Explorer on my Windows 2000 PC and typed in a URL. To my surprise, I connected immediately. I decided to do a preliminary test of my security by browsing to Steve Gibson's web site: http://grc.com. Steve Gibson is considered an expert on Internet security by many of the people I read comments from on Google. At Steve's web site there is a web based application called 'Shields Up!' The application tests your PC for Internet vulnerabilities by probing for open network ports over the Internet. Mr. Gibson states right on the first page of the application that his test is not meant for professional testing of business class firewalls, but it is a good start for the home user. The test came back good. Even though I was running the test from My Windows 2000 PC, none of the ports I had open showed up open. This is what I was shooting for. I wanted my Windows 2000 PC to have some open network ports. This way I can share files between my home PC's without opening myself to attack.

After I confirmed my firewall was working, I ran **nmap** (http://www.insecure.org/nmap) against my original setup and my new setup. Nmap first scans for open ports, then measures security by how difficult it would be to break in to the host over a network connection. A score is displayed every time nmap scans a host. Higher scores represent a more secure system. My initial scores with Windows 2000 Internet Connection Sharing installed was around 3500. With Linux sharing my Internet connection and the firewall running, I get scores near 3,000,000. Here are the results s listed by nmap:

Windows:

```
Starting nmap V. 2.54BETA7 (www.insecure.org/nmap/)
Interesting ports on  (bon):
(The 1528 ports scanned but not shown below are in state: closed)
Port       State      Service
135/tcp    open       loc-srv
139/tcp    open       netbios-ssn
1002/tcp   open       unknown
1026/tcp   open       nterm

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=353
No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
```

SInfo(V=2.54BETA7%P=i386-redhat-linux-
gnu%D=6/16%Time=3B2B79EA%O=21%C=1)
TSeq(Class=RI%gcd=1%SI=209A)
TSeq(Class=RI%gcd=1%SI=3291)
TSeq(Class=RI%gcd=1%SI=220E)
T1(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E
%ULEN=134%DAT=E)

Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds

Linux

Starting nmap V. 2.54BETA7 (www.insecure.org/nmap/)
Interesting ports on (malcom):
(The 1527 ports scanned but not shown below are in state: closed)
Port State Service
22/tcp open ssh
3001/tcp open nessusd

TCP Sequence Prediction: Class=random positive increments
Difficulty=2664278 (Good luck!)
No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
TCP/IP fingerprint:
SInfo(V=2.54BETA7%P=i386-redhat-linux-
gnu%D=6/16%Time=3B2B7ACD%O=22%C=1)
TSeq(Class=RI%gcd=1%SI=514E17)
TSeq(Class=RI%gcd=1%SI=514E78)
TSeq(Class=RI%gcd=2%SI=28A756)
T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=Y%TOS=C0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=

E%ULEN=134%DAT=E)

Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds

Since I installed the firewall, I also started watching my log files. I frequently find IP addresses in my /var/messages/secure log file which were denied access by my firewall. By running the command traceroute, I have been checking to see where these IP originate. I recently found a tool called 'Visual Route.' This tool is not freeware. It costs about $40.00 for a single user running on a Windows platform. This tool is great for tracing IP addresses. It actually gives you a map of the world and shows you (by drawing lines) where an IP originated from and every hop between you and the origin. Visual Route can be downloaded from: http://www.visualware.com/visualroute/index.html. It is very interesting finding out where the would-be attackers are coming from. I don't try to retaliate or attack these people's machines. Just the satisfaction in knowing my firewall is doing it's job is good enough for me.

The only real downside to my configuration that I can see is the fact that I cannot use the Internet without at least two PC's running. As stated earlier in this document, this is my home PC. I rarely use my Linux PC for anything besides the firewall. The other route I could have taken would be securing my Windows 2000 PC and just connecting my DSL modem directly to it. Many changes would need to be made to my Windows 2000 configuration.

Lately I have been reading good reviews on a free firewall call Zonealarm. It can be downloaded from http://www.zonealarm.com. I thought I would try it out. This free software is basically an automatic version of the firewall I am running in Linux. Once installed, Zonealarm automatically blocks most types of attacks (I wouldn't say all of the attacks out there, there are too many and new ones every day). Also, when an 'Internet Aware' application is launched, Zonealarm will prompt the user to authorize the connection the application is trying to make. The first time I launched Internet Explorer with the firewall running, a message prompted my to authorize the connection. Since I want to browse the web with this browser, I clicked OK and I asked the software not to prompt me anymore regarding Internet Explorer. The same was true of Outlook, Windows Media Player, and a couple of other apps. This type of protection is relevant because there are many type of software that can be downloaded for free and installed by a user. This software may seem like a fun way to get music or a game. Once the application is running, it can make connections to the Internet and allow others into your files and O/S without you ever knowing. Zonealarm also automatically blocks incoming netbios requests. This will keep all of your Microsoft networking information private. Attackers love netbios information. When a PC running windows with the default network setup connects to the Internet, attackers can enumerate all of your PC's shares and in Windows 2000, an attacker can usually get the username of the user currently logged in, getting this information is the start of an attack. The free version of Zonealarm

will only protect one PC. If you want to share your Internet connection, you have to pay $19.99 for the professional version. With the professional version and two network cards, I shared my DSL modem with the rest of my PC's just like my Linux configuration listed earlier. Zonealarm logs IP addresses of would be attackers and will even help you trace them back to an ISP or registered domain. Both the Zonealarm and the Linux based configurations seem to cover my needs adequately. I have chosen to keep both. I can switch between the two configurations with minimal hassle.

Appendix A

```
#########################################################################
############
#  /etc/seawall.conf V4.1 - Change the following variables to match your setup
#
#  WARNING: Do NOT comment out any of the definitions in this file; if you
#         don't need the definition, make it an empty string (e.g., var="")
#
#  This program is under GPL [http://www.gnu.org/copyleft/gpl.htm]
#
#  This file should be placed in /etc.
#
#  (c) 1999, 2000, 2001 - Tom Eastep (teastep@evergo.net)
#########################################################################
############
# SECTION 1.
#
# All users must customize this section
#########################################################################
############
# Interface through which you connect to the internet

internet="eth0"

# If your ISP has assigned you a static IP address, enter it here. If you have a
# dynamic IP address, leave this null. If you have a dynamic IP, your network
# interface must be started before you start the firewall. If you use PPTP to
# interface to the internet, set myip="pptp" or myip="PPTP". If you use PPP, PPPoE
# or some other interface type that doesn have a static IP but doesn't use
# DHCP either set myip="ppp" or myip="PPP".

myip=""

# Interface(s) for your local network (Optional - if you're running Seawall
# on a stand-alone system, leave this blank ).
#
# IMPORTANT: your local interface(s) must be up and configured before you start
#          Seawall
#
# Examples:
#         local="eth1"
#         local="eth1 eth2"

local=""
```

```
# Firewall Strength -- Set this to "Yes" or "yes" if you want to use the
#               /etc/seawall/apps file to specify the TCP client
#               applications that can run on the firewall system.
#
#               Set to "No" or "no" if you want to be able to run
#               Any TCP application on the firewall system.
#
#               If this variable is left empty (strong="") then
#               if 'local' or 'dmz' is non-empty then 'strong=Yes'
#               is assumed otherwise 'strong=No' is assumed.

strong=""

# Non-forwarded interfaces -- Set this variable to a list of interfaces
#                   that you want open from this system but
#                   that will not be forwarded to any
#                   other interfaces
#
#                   This variable differs from 'nonmasq' below
#                   in that 'nonmasq' interfaces are forwarded
#                   to/from local interfaces whereas 'noforward'
#                   interfaces are not
#
#                   Example: noforward="ppp1"

noforward=""

# Non-forwarded subnetworks/addresses -- Set this variable to a list
#                       of addresses/subnetworks that you
#                       don't want forwarded locally. This
#                       variable is primarily used when
#                       you have IPSec tunnels on your
#                       gateway and you want to include
#                       additional addresses/subnetworks in
#                       the firewall's spoof protection.

noforwardnets=""
```

```
# Dial In PPP -- If you support PPP dial-in access to your gateway system,
#             list the remote IP addresses that you assign to these
#             sessions. Multiple entries are allowed and entries may
#             be subnets or individual IP addresses:
#
#             Example: dialinppp="192.168.10.5 192.168.10.16/28"

dialinppp=""


# Local Port Range -- By default, Linux uses the range 1024:4999 for
#                 dynamic local ports. This range can conflict
#                 with applications using registered ports in
#                 this range. The IANA recommends that the
#                 local port range 49152 or higher. If you want
#                 to specify a different range, you may do so in
#                 this variable
#
#                 Example: localports="49152:53999
#
#   WARNING: If you change the value of this variable, you must
#          restart all inet applications running on your system.

localports=""

# Set this to "[Yy]es" if you wish to log denied packets (Recommended)
#
log="Yes"

# Set this to the name of the lock file expected by your init scripts. For
# RH6.*, this should be /var/lock/subsys/firewall. On Debian and LRP, it
# should be /var/state/firewall. If your init scripts don't
# use lock files, set this to "".
#

lockfile=/var/lock/subsys/firewall

# NTP Servers (optional) -- set this if you need to accept messages ntp
#                       servers
#
# Example:
#         ntpservers="203.191.149.12"
```

```
ntpservers=""

#
# NTP Uses non-privileged ports (optional)
#
#                    -- if the ntp clients that you run on your
#                       firewall use non-privildged local ports,
#                       set this varilable to "[Yy]es" (example:
#                       ntpnonpriv="Yes" ).

ntpnonpriv="No"


# DNS Servers (optional) -- If you use forwarding DNS servers or servers at
#                       your ISP, list them here -- if this variable is
#                       empty, the firewall will accept UDP port 53
#                       packets and TCP port 53 non-SYN packets from all
#                       sources. If you set the variable to "[Nn]one",
#                       the firewall will only accept masquerade DNS
#                       replies (if you have your own DNS server
#                       behind the firewall).
#
# Examples:
#         dnsservers="203.191.149.10 203.191.150.12"
#         dnsservers="None"

dnsservers=""

# DNS Local Ports (optional) -- This variable contains the range of local ports
#                         used by DNS clients. Normally this will be
#                         the "localports" range (which will be assumed if
#                         you leave this variable empty).
#
#                         If you run dnscache on your firewall box, you'll
#                         want to set this to:
#
#                            dnslocalports="1025:"

dnslocalports=""

# ICQ TCP Ports (optional) -  On a standalone system, you will need to configure
#                         your ICQ client "Behind a firewall" and specify a
#                         range of ports for it to use for incomming connections.
#                         That range of ports should also be specified in this
#                         variable (example: icqports=3000:3999).
```

```
#
#                   If you use the ICQ Masquerade module from
#                   http://members.xoom.com/djsf/masq-icq" and you
#                   override the default ports used for connection
#                   forwarding (60200:61000) then specify your
#                   port range in this variable.
#


#                   If you use the ICQ Masquerade module but do
#                   not override the default ports, then just include
#                   "icq" in the 'modules' variable below.
#
#                   If you are using the module, Seawall will also accept
#                   SYN packets on ports 61000:65095 -- THIS IS A
#                   POTENTIAL SECURITY HOLE but is required to
#                   make ICQ chat and file transfer work properly
#
# Example:
#       icqports="60200:61000"

icqports=""

# If you are running Seawall on a single system (no Masquerade), you're
# finished.

#########################################################################
############
# SECTION 2.
#
# If you don't need to masquerade PPTP clients or a PPTP server and you
# don't run the pptp client or PoPToP on your firewall, go to section 3
#########################################################################
############
#
# If you select any of the following, you must add GRE (protcol 47) to
# your /etc/protocols file as follows (delete leading "#"):
#
#       gre     47      GRE   # Generalized Routing Encapsulation

# External PPTP Servers -- Set if you masquerade MS PPTP ; you will also need John
#                   Hardin's PPTP masquerade patch from
#                   http://www.wolfenet.com/~jhardin and
#                   you must add GRE (protocol 47) to /etc/protocols
#                   This variable should list the EXTERNAL PPTP servers
```

```
#                       that your clients wish to connect to
#

pptpservers=""

# Internal PPTP Server -- Set this if you have a PPTP server behind your firewall.
#                     Do NOT set this variable if you run PoPToP on your firewall.
#
#   You will also need to
#
#   -  Install John Hardin's PPTP masquerade patch from
http://www.wolfenet.com/~jhardin
#   -  Install ipfwd from  http://www.pdos.lcs.mit.edu/~cananian/Projects/IPfwd
#   -  Install ipmasqadm from http://juanjox.kernelnotes.org
#

pptpserver=""

# External IPSEC Servers -- Set if you masquerade IPSEC clients; you will also
#                       need John Hardin's IPSEC masquerade patch from
#                               http://www.impsec.org/linux/masquerade/ip_masq_vpn.html.
#
#                               This variable should list the EXTERNAL IPSEC servers
#                       that your clients wish to connect to

ipsecservers=""

# PoPToP on your Gateway -- If you run PoPToP on your gateway, then:
#
#                       o In /etc/ppp/options, include "proxyarp"
#                       o In /etc/pptpd.conf, set localip to the address
#                         of one of your local interfaces and set
#                       remoteip to a set of unused IP addresses in the
#                       subnet for that interface.
#                       o Specify the name of that interface in this
#                         variable (Example: poptop="eth2"

poptop=""

# External PPTP Clients -- If you would like to restrict who may connect to your
#                       PPTP server, list the hosts/networks here
#                       if this variable is empty, anyone may connect
#

pptpclients=""
```

```
# PPTP Client -- If you run the PPTP client on your firewall system, set
#               this variable to "Yes" or "yes".

pptpclient="No"

########################################################################
############
# SECTION 3.
#
# Only usrs that need routing need to customize this section
########################################################################
############
# DMZ (Optional)       If you want to configure a DMZ, it must be interfaced
#                  to your firewall by it's own interface. Place the
#                  name of the interface in this variable (Example:
#                  dmz="eth2"). Do not list this interface in the "local"
#                  variable above.

dmz=""

# Hidden Subnetworks -- If you want to masquerade subnetworks that are not
#  (Optional)              directly connected to the firewall via one of the
#                  interfaces listed in $local, you may describe them
#                  here (e.g., localnets="192.168.12.0/24"). Most people
#                  can just leave this empty

localnets=""

# Non-masqueraded interfaces -- Set this variable to a list of interfaces
#                       that you want open from this system but
#                       that will not have internet access.
#
#                       Example: nonmasq="eth1"

nonmasq=""

# Non-masqueraded networks -- Set this variable to a list of subnetworks
#                      and/or addresses that you want routed
#                      locally but do not want to have internet
#                              access.
#

nonmasqnets=""
```

```
# POP Server (optional) -- set this if you need to access POP3 servers
#                       from your firewall

popservers=""

# SMTP Server (optional) -- set this if you need to access SMTP servers
#                        from your firewall

smtpservers=""

# IPMASQ Modules -- List the "ip_masq_*" modules that you want Seawall to load
#                when it starts. Note to LRP users: this should be left empty
#                when running Seawall under LRP
#
# Example:
#         modules="ftp raudio icq"
#
modules=""

# MASQ timeouts
#
#   2 hrs timeout for TCP session timeouts
#  10 sec timeout for traffic after the TCP/IP "FIN" packet is received
# 160 sec timeout for UDP traffic (Important for ICQ Masq'ing)
#
# would use the command: masq_timeouts="7200 10 160"

masq_timeouts=""
```

Questions:

1. The default or 'out of the box' configuration of Windows 2000 Professional is secure enough for use on the Internet.

    A. True
    B. False

2. Most ISP's who offer broadband connections to the Internet also offer protection from outside attacks no matter what O/S you run.

    A. True
    B. False

3. Linux can be used as a firewall.

    A. True
    B. False

4. There are many 3d party firewall applications that can be purchased to protect your PC from outside attacks.

    A. True
    B. False

5. RedHat Linux 7.0 ships with a firewall pre-configured for use on a home network.

    A. True
    B. False

6. Nmap is a utility used for:

    A. Configuring Windows 2000 web services.
    B. Scanning for open tcp/udp ports on any networked computer.
    C. Cleaning toilets.
    D. Drawing a map of the United States.

7. Shutting down unused network services on networked computers helps to:

    A. Keep you from getting work done.
    B. Connect other computers to your computer from outside networks
    C. Keep the risk of vulnerabilities & successful attacks down.
    D. Save on power costs.

8. One way to find out if anyone has been trying to break into your networked computer is:

    A. Unplug you network connection and see if anyone complains.
    B. Call your ISP.
    C. Check your log files and look for any unusual activities.
    D. Install a firewall.

9. Frequently checking for updates to your operating system:

    A. Keeps you informed of new games.
    B. Is a good way to get new wallpaper and screensavers.
    C. Will keep you up to date on new products and beta software.
    D. Will help determine if your networked computer is vulnerable to attack

10. 'Hackers' or malicious Internet users usually try to break into home computers:

    A. To use the compromised computer for their own purposes.
    B. To gain notoriety.
    C. To get access to credit card information.
    D. Just to say 'hello' in some electronic way.

Answers:

1. False, the default configuration of Windows 2000 is not secure. Configuration changes need to be made.
2. False, while some ISP's will watch for some attacks, most do not.
3. True, there are many different ways to use Linux as a firewall.
4. True, there are many 3d party add-ons to Linux which provide firewall services.
5. False, RedHat has packaged a firewall into the newer release (7.1).
6. B. Nmap is used to scan network ports
7. C. Shutting down unused network services greatly reduces the risk of attack/compromise.
8. C. Most operating systems these days have a logging utility that keeps track of all activity on the computer. These logs can provide all of the information you need to find out what (or who) has been on your computer.

9. D. Most operating systems these days have web sites associated with them and list new vulnerabilities as they come up.
10. A. Most, (not all) attacks of home computers are related to the use of the compromised computer as a server for some type of file transfer orchestrated attack on another computer.