Global Information Assurance Certification Paper

Securing Linux Installations

This paper describes what a new Linux installation should look like. Due to the nature of our organization's culture and rules, we can not unilaterally force users to apply our suggestions. Therefore, we look for and stress improvements that can be easily and quickly be made. Such improvements will have a greater likelihood of being implemented and is our best route to increasing security.

The workstations that exist on our private network are primarily used to run general-purpose productivity applications and scientific computing. The general-purpose applications are word processors, spreadsheets, e-mail clients and web browsers. The computing users or programmer's utilize applications such as Fortran, C, C++ and various libraries. Nearly all workstations run either Red Hat Linux or Sun Microsystems Solaris.

We break our suggested configuration modifications into 4 categories:

1. File system layout
2. Package selection
3. Removing unnecessary daemons
4. Constructing a simple firewall

The first three steps are based on SANS security configuration guidelines. The last step was derived from one of our engineer's recommendations.

File system layout

Many employees use the basic Red Hat "workstation" file system layout. By default it formats two file systems: root (/) and home (/home). While this is generally adequate, we suggest using two additional file systems: /usr and /var. The benefit is two-fold: first, we can protect the /usr file system from unauthorized writing by making it read-only; second, separating the /var directory structure from the root (/) file system minimizes the danger of a successful DOS attack filling up root.

Modern computers have enough space to provide generous allocations for the file systems. We recommend a minimum of 2 GB for /usr and 1 GB for /var whenever possible; the generally don't have to be much bigger, however. With the /usr and /var directories using their own file system, root (/) can easily run with 100 MB of space. Finally, the /home partition is given the rest of the disk space minus the swap partition. Note that we prefer to link the optional software directories /usr/local and /opt to a directory on /home. This is optional, but has the advantage of pooling space in an efficient manner.

Package selection

The next step in securing a workstation is to minimize the number installed packages while still meeting the user's needs. The general user need spreadsheets, word processors, an e-mail client, and web browser. However they would not need NFS,

HTTP servers, GCC or ECGS. The programers need Fortran, C, C++, compilers, LaTech, mat lab, Ghostscript and TEX. Both users tend to want multimedia players, simple graphic utilities, etc. However SMTP, Samba, NFS servers, DNS servers or FTP servers are not necessary.

We recommend to our non-programers that they do not install the following packages:

Communication applications - efax, irk, irk-help, lrzsz, minicom and all modem related packages

Editors - emacs-el, jed, joe vim-X11, vim-color, vim-comon, vim-enhanced, vim-minimal

Emulators - dosemu, dosemu-freedos, xdosemu

Graphics - zgv

Mail - elm, sxmh, fetchmail, mailx, metamail, mutt, nmh, pine, send mail

Networking - arpwatch, libpcap, lynx, ncftp, rsync, tcpdump, wget

News - slm, tin, trn

Publishing - groff, groff-gxditview, lout, loutdoc, sgml-tools, tetex, tetex-afm, tetex-doc, tetex-dvilj, tetex-dvips, tetex-xdvi,texinfo

Kernel - kernel source

Daemons - sendmail-cf, sendmail-doc, uucp

Languages - All languages except Perl

Libraries - All development libraries

Tools - fles, gettext, gperf, indent, xwpe

Games - Games do not need to be installed

Networking - Admin - anonftp, caching-nameserver, nfs-server-clients

Daemons - apache, apache-devel, bind, clenfeed, dhcp, dhcpcd, gated, inn, mars-new, mod_perl, mod_php, mod_php3, nfs-server, sendmail, squid, suid-novm, ucd-snmp, wu-ftpd

Development - bind-devel

Utilites - bind-utils, comanche, dip, fwhois, mgetty-sendfax, mgetty-voice, nc, ncpfs,yp-tools, biff, finger, ntalk, pidentd, rsh, rusers, rwall, rwho, samba, telnet, tftp, ypbind, ytalk

Programmer's should also not install the packages listed above. However, they will want to install the packages found in the development, development library and languages groups as appropriate. They will also want to install their editors of choice from

Note that many of the communication applications are modem related. Unauthorized use of modems can bypass our firewall and is discouraged. By eliminating the modem related communication programs, this problem can be minimized. It would also circumvent the firewall and already installed security.

Mail is another service that poses risk from a security viewpoint. Mail is susceptible to intrusion and is hard to individually maintain security. Currently, we have a client-server corporate e-mail system where we the Netscape e-mail client. This system is secure and simple to use. News services are not needed on the machines because of

simple retreival through Netscape Newsreader. Under networking administration we specifically would eliminate anonftp to prohibit users from obtaining files from the machines without having and account. Included in the network administration the nfs-server is not needed to mount network attached file systems from a server thus eliminating the ability to access other machines if the system is hacked.

Stopping unecessary daemons

Unnecessary daemons pose another easily solved security problem.

- Web servers such as Apache that offer access to the web are not included on the workstation.
- bind is used in DNS, workstations do not need to be a DNS servers because other servers support this function. The configuration file caching-nameserver is a bind file that is not necessary.
- DHCP provides a server and a relay agent that is not supported on the current workstation.
- NIS and yp-tools applications are not included because the current workstations do not utilize them.
- A simple protocol that allows users to look up information on other machines is referred to as finger. The elimination of finger prohibits users from obtaining other's ip addresses and user information.
- The internet talk protocol known as ntalk opens ports that can be hijacked, again providing an additional security breach.
- Rsh, rlogin, and rcp are a suite of programs that allow users to run commands on remote machines, login to other machines and copy files between machines. Remaining consistent with our access the rsh, rlogin and rcp programs have security holes. The program rusers allow users to find out who has logged into various machines, by abolishing rusers and rwho it would disable people to find names of users to hack passwords. Instead ssh should be used because it uses encryption and can authenticate both the user and originating host.

Samba provides a SMB server used to provide network services to users of other operating systems if this protocol is needed it should be run exclusively on a server not individual workstations.

Adding a simple firewall

Finally, we add a simple IP filtering firewall to increase security. Any hacker who gains access to your network will be able to attack individual machines unless further measures are taken. This is why adding a firewall is a good idea. By adding IP packet filters on each computer, every computer on the private network becomes its own firewall becoming a second layer of defense forcing an intruder to defeat two levels instead of just one.

Redhat 7 comes with ipchains as an RPM package, is easily installed and configured. IP filters are very simple to set up when a computer is used mostly to access external resources. If your workstation does not share files or services to another computer, then a very simple rule set can be used.

The handful of rules shown below configures a Linux computer for mostly one-way connection to the network. IP packets can go out but only their return packets can come back in. (Actually, this isn't quite true. Only a stateful filter can do that but these rules approximate that ability). For instance, when you click on a remote web page, our browser sends out an http packet (port 80) to the remote web server. The remote server sends back information in the form of http packets (on arbitrary ports above 1024).

Listing 1: Simple IP filtering rules to protect a workstation

```
# Rule 1
# Start by denying all packets in all directions. Your firewall is perfect. No one can get
in - or out!
ipchains - P input DENY
ipchains - P output DENY
ipchains - P forward DENY
# Rule 2
# Back off and allow all internal network traffic on the loopback (logical) interface
ipchains - A input -I lo -j ACCEPT
ipchains -A output -I lo -j ACCEPT
# Rule 3
# Back off more and allow all outgoing TCP/UDP and ICMP packets
ipchains -A output -j ACCEPT -I eth0
# Rule 4
# Allow the returning TCP packets back in through your firewall
ipchains -A input -p TCP -j ACCEPT -I eth0! -y
# Rule 5
# Allow DNS UDP packets in (DNS is necessary for converting IP Address names into
numbers)
ipchains -A input -p UDP -j ACCEPT-I eth0 -s 0.0.0.0/0  domain
```

That's all that's necessary to create a simple, but effective firewall. The script works as follows:

1. Deny all network on all interfaces. You now have the perfect firewall. You can't use your computer for anything, but you are safe.
2. Open up the loopback (lo) interface to all network traffic. The loopback is a logical network interface - it is not a physical interface - used for internal network traffic and must not be blocked in any way. Open it up and your computer can speak to itself.
3. Allow all out going packets. Any TCP/UDP and ICMP packet can go out of your NIC.
4. Allow TCP packets with their "ACK" bit set to return through the firewall. The "ACK" bit, in the IP packet header, indicates that the packet is in acknowledgement to an outgoing TCP packet. This rule allows two-way TCP communication. For instance, when you click on a web page somewhere, your browser sends out a packet. The remote server sends back packets with their "ACK" bit set. This rule permits you to see the information that is returned by the server.

5. Finally, by allowing UDP packets from the domain port (53), name server requests can be made. Domain Name Service (DNS) is used to convert IP names to numeric addresses. For instance, www.theonion.com translates to the address 216.165.161.17.

This firewall permits you to go out to your private network, the Internet and any other networks that you may be connected and have at it. It also prevents most every direct or indirect attack.

To set up the firewall do as follows:

1. login on the Linux computer as root.
2. copy the ipchains rules into a script such as /tmp/ipfilter.rules.
3. Make the script executable.
   chmod o+x /tmp/ipfilter.rules
4. Run the script.
   /tmp/ipfilter.rules
5. Run the ipchains utility ipchains-save and direct its output to /etc/sysconf/ipchains.
   ipchains-save > /etc/sysconfig/ipchains

You can look at the rules by running the command:

ipchains -L

You should see the rules listed for you.

Allowing Secure Shell (and other services) into your Workstation

Well, you might not like the one-way restrictions on your workstation. The rules can be loosened just a bit to permit Secure Shell in. Secure Shell is a commercial product that provides an interactive connection from one machined to another that is encrypted. Secure Shell provides both interactive communication as well as file transfers (sftp) and remote command execution. It can also forward X and other ports. It is the Swiss Army knife of encrypted communications.

The following rule allows ssh into your workstation.

ipchains -A input -] TCP -j ACCEPT -I eth0 -s 0.0.0.0/0 ssh

In this case the hole is opened only to Secure Shell, which in it is a kind of firewall. Secure Shell works at the application level and provides its own authentication. It is beyond the scope of this article to describe Secure Shell configuration, but it can be set up so that both the user and host are authenticated before access is allowed. Thus, even if it is known that you have an opening on port 22 an intruder cannot just waltz in. Above is an example of a simple installation and how to secure a Linux workstation within the parameters in which I have to work on my network given the tools that I have at hand.

Conclusion

We have found the security measures described here to be useful within our network. They provide easy or relatively easy steps that our user base does not object to. There are certainly other steps that we can take and with time we expect to expand our suggested configuration whenever possible.

Roger A. Retallack

References:

http:\\www.sans.org\topten.htm
http:\\www.redhat.com
http:\\netfilter.filewatcher.org/ipchains
http:\\www.flounder.net/ipchains/ipchains-howto.html