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.

Build your own firewall using SuSE Linux: A mechanics guide.

There are several hundred competing router and firewall solutions available today on the market. Different types are available that are considered low end for the home user and high end for corporate networks and everything else in between. Services and throughput requirements may differ between these different networks but in regards to security is there any difference? Today the security requirements could be different but tomorrow the security implementations and methods may be the same. Let's assume for today we ne...

Copyright SANS Institute
Author Retains Full Rights

AD

Paul O'Neil
Version 2.5b (revised February 2003)
Build your own firewall using SuSE Linux: A mechanics guide.

Abstract

There are several hundred competing router and firewall solutions available today on the market. Different types are available that are considered low end for the home user and high end for corporate networks and everything else in between. Services and throughput requirements may differ between these different networks but in regards to security is there any difference? Today the security requirements could be different but tomorrow the security implementations and methods may be the same. Let's assume for today we need a low cost yet highly functional router and firewall combination for a small office or medium sized business that offers Internet services such as email to its small network of employees and web services such as a web server serving web pages to its customers. The following paper describes the different tools that can be used in setting up an appropriate router and firewall combination using Linux that offers the necessary functionality and security to its users as well as the means to monitor it by an administrator. What follows is not a how-to but an outline to take "reasonable care" following due diligence in providing network security in an environment that requires it.

Given that "predictive monitoring of Unix systems is so expensive, in sysadmin time, in education, in raw computational power" (Annis) we should desire to monitor our router and firewall combination with tools that target main areas of concern inside a Linux system. We need a flexible distribution that can accommodate a potentially wide range of open source tools and a system setup that is not overly complex but technically sophisticated enough to mitigate problems that may arise. A managed process needs to be implemented that vigilantly monitors and documents the prevalent activity inside the system. The design of this system is built with the philosophy that "security is a process, not a product" (Schneier, p.273). This paper is an attempt to put that sentiment into practice. Realizing there is no silver bullet when it comes to network security, this paper can act as a guide for anyone with intermediate level experience using Linux systems.

"Security processes are not a replacement for products; they're a way of using security products effectively. They can help mitigate the risks. Network security products will have flaws; processes are necessary to catch attackers exploiting those flaws, and to fix the flaws once they become public. Insider attacks will occur; processes are necessary to detect the attacks, repair the damages, and prosecute the attackers. Large systemwide flaws will compromise entire products and services (think digital cell phones, Microsoft Windows NT password protocols, or DVD); processes are necessary to recover from the compromise and stay in business" (Schneier)

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

SETUP

The requirements for this system rely on four core principles as outlined by Bruce Schneider when stating “the military breaks the process down into four generic steps” in Secrets and Lies, Digital Security in a Networked World, p 376.
The design incorporates the following: Detection, Localization, Identification, and Assessment.

We must have tools installed that allow us to know we are under attack or that potentially malicious activity has occurred. Detection tools should be installed to indicate such activity. Localization tools are necessary after detection. We must have the ability to measure on what part of the system the activity has occurred. The ability to identify is important for a few reasons. One is that if the attacker can be identified then maybe legal recourse can be taken but at least other tools can used to prevent the source of the malicious activity from accessing the system. Finally we must be able to assess and understand the activity itself and the damage if any that has occurred.

List of tools used in building your own firewall using SuSE Linux: A mechanics guide: with brief descriptions taken from the respective sites.

Linux Distribution: SuSE Linux Professional 8.1.
The Linux kernel using 2.4.x. Recompiled with only required services for a more efficient kernel.
Iptables: The netfilter/iptables project is the Linux 2.4.x / 2.5.x firewalling subsystem.It delivers you the functionality of packet filtering (stateless or stateful), all different kinds of NAT (Network Address Translation) and packet mangling.
Fwlogwatch: Produces Linux ipchains, Linux netfilter/iptables, Solaris/BSD/Irix/HP-UX ipfilter, Cisco IOS, Cisco PIX, NetScreen, Windows XP firewall, Elsa Lancom router and snort IDS log summary reports in plain text and HTML form and has a lot of options to analyze and display relevant patterns. It can produce customizable incident reports and send them to abuse contacts at offending sites or CERTs. Finally, it can also run as daemon (with web interface) doing realtime log monitoring and reporting anomalies or starting attack countermeasures.
Harden_suse: harden_suse is system security script for SuSE Linux only. It makes several changes to the system configuration to make the operating system very secure and therefore very resistent to local as to remote attacks.
Portsentry: PortSentry is a program designed to detect and respond to port scans against a target host in real-time.
Nabou: Nabou is a system integrity monitor.
Perro: It is a set of three daemons that logs incoming IP/TCP, IP/UDP and IP/ICMP packets

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Seccheck: The SuSE Security Checker is a security tool which does a security analysis on a daily, weekly and monthly basis.
Chkrootkit: Is a tool to locally check for signs of a rootkit.
VPND: The virtual private network daemon vpnd is a daemon which connects two networks on network level either via TCP/IP or a (virtual) leased line attached to a serial interface. All data transfered between the two networks are encrypted using the unpatented free Blowfish encryption algorithm. .
IPlog: iplog is a TCP/IP traffic logger.
OpenSSH is a FREE version of the SSH protocol suite of network connectivity tools that increasing numbers of people on the Internet are coming to rely on. Many users of telnet, rlogin, ftp, and other such programs might not realize that their password is transmitted across the Internet unencrypted, but it is. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks.
SuSEfirewall2 is a highly secure but easy to configure firewalling/packet filtering tools for Linux for the 2.4 Kernel.
SNARE is an audit subsystem for the Linux open source operating system.
NTP is a protocol designed to synchronize the clocks of computers over a network. NTP version 3 is an internet draft standard, formalized in RFC 1305. NTP version 4 is a significant revision of the NTP standard, and is the current development version, but has not been formalized in an RFC.

The network environment we will be securing is as follows: One computer will serve as a firewall in a network setup based on the “Three-Homed Firewall” DMZ architecture as described by Michael Bauer in Building Secure Servers with Linux, p.25. This computer will have three network interfaces. One interface will be the edge interface connected to the public network. The second interface is connected to a DMZ or screened subnet. The third is connected to the private or internal network.

The DMZ contains one web server hosting a company’s website. The internal network contains at least one client computer.

All tools are open source and freely available from the Internet. Initially we need to install the particular Linux version. SuSE Linux was chosen primarily due to my familiarity with it but any flavor is acceptable. An intermediate understanding of installation, setup and use of Linux system administration is expected since I will not go over step-by-step installation of any tool described here. There are a few basic guidelines to mention for the install. Primarily, the setup should install as few services and programs as possible. Preferably a static kernel with no loadable module support enabled. This setup would be for the advanced user and may cause problems with the tools mentioned in this paper. For our purposes a stock kernel is sufficient. We will need the capability to compile source code in situations the binaries won’t work. There are tools where corresponding libraries will be needed so at least for those described here they can be installed as they are needed for each tool.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

SuSE Linux comes with a nice tool that sets up firewall rules for you based on the criteria specified in a setup script. Although a basic understanding of Iptables is recommended it is not required here. However, learning how to edit, configure, and run the script is a necessity since many changes can occur during normal usage.

The firewall script offered by SuSE is named SuSEfirewall2. We will not be attempting to achieve security through obscurity in our approach. To accommodate our network requirements a description of firewall rules and its setup is as follows: After interface definitions are set such as DMZ, internal and external interfaces, routing between the these interfaces is turned on which allows masquerading for the internal machines. Care is taken to be sure which machines are allowed to masquerade at this point. A nice feature in this script is the specificity with which you can define what is allowed such as a target network, protocol and service which is appended to the host such as: 192.168.10.0/24,0/0,tcp,80 that allows the 192.168.10.0 network to use www to the Internet. Next the script can be set to “protect” the firewall from the internal network which can prevent attacks on services that are running on it. The only services we will be offering at this point on the firewall that are in other words reachable from the Internet are SSH on port 22, Rsync on port 873 and SNTP on port 123. Of course only port 80 traffic or www will be allowed to the DMZ from the external interface. Another useful feature is a setting from trusted hosts. Here you can define IP numbers that are allowed to access certain services however, authentication for those services should of course exist but for our purposes we can leave this blank. Next we will redirect access to port 80 to another local port on the firewall that is running Squid Proxy as follows “192.168.10.0/24,0/0,tcp,80,3128”. To be sure Squid should be running on another system but it will run on the firewall to be used for logging of traffic and the control it provides. Other features are available and the defaults should be used as well as the extra kernel TCP/IP security features. Rsync on port 873 will be used for backing up files and SNTP on port 123 will be used for our timeserver service. Keeping accurate time is essential for good monitoring.

Basically are setup follows a strict rule set and allows only traffic that is explicitly defined.

To further strengthen our firewall from remote and local attacks we will use the harden_suse script. This script makes several advanced changes and an intermediate level of understanding a Linux system is required before executing it. A brief description take from http://www.suse.com/~marc/harden_suse.html is here:
1) deactivate all network services, except very few security services (e.g. SSH, Firewall, VPN)
2) change the filer permissions to a secure state

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3) comment out all services in /etc/inetd.conf and secure the tcpwrapper to allow only localhost access
4) secure the login process (log all login attempts, show last/failed logins, root login only from console)
5) secure the passwords (long passwords enforced, password change after 40 days, weak password warning)
6) strong permissions on /home directories of users and a strict umask (077) for all users
7) secure configuration of SSH/SSHD (disables/enables options for better security)
8) remove privilege of all unknown suid files on the system
9) remove world write permissions on all unknown world writeable files on the system
10) show legal disclaimer in the login banner, motd and lilo boot menu.

After running this script some changes in /etc/hosts.allow, etc… will be required to accommodate network administration as care is required after this point so as to not lock out the network administrator from the system. It assumed that configuration to the system is being accomplished locally, however.

OpenSSH should be properly configured at this stage. The decision to use keys for remote administration is up to the administrator but for flexibility I would recommend password authentication since adequate monitoring and detection will be implemented later. Trust at this point is a philosophical question. I may not place as much trust in any services I run as I should be placing in my golf swing but I will vigilantly monitor the services I do offer and trust them as much as is feasible.

Nabou is an interesting tool available from http://www.nabou.org and is a system integrity monitor. While not as effective and scalable as tripwire it uses MD5 and other comparatively capable technologies that offers at least baseline of measurement for tampering that is easily configurable and not overly complex. Nabou can be considered a low scale intrusion detection system but will be used for basic system monitoring in this case. It will look for any changes to user accounts etc… and system files using the MD5 check. Any inconsistency can be configured to alert the administrator by email if desired. It should be configured to run nightly that generates a report of its analysis. Nabou is the first tool in our arsenal that falls under the “detection” category to adhere to the first principle in our system.

Security Check is another script that is easily added to the system that augments the Nabou system checks. It runs daily, weekly and monthly and is streamlined to email a report if any inconsistency is detected. Taken from http://www.suse.com/~marc/README.seccheck a description follows:

/etc/passwd check : length/number/contents of fields, accounts with same uid

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

accounts with uid/gid of 0 or 1 beside root and bin
/etc/shadow check : length/number/contents of fields, accounts with no password
/etc/group check : length/number/contents of fields
user root checks : secure umask and PATH
/etc/ftpusers : checks if important system users are put there
/etc/aliases : checks for mail aliases which execute programs
.rhosts check : checks if users' .rhosts file contain + signs
homedirectory : checks if homedirectories are writable or owned by someone else
dot-files check : checks many dot-files in the homedirectories if they are writable or owned by someone else
mailbox check : checks if user mailboxes are owned by user and unreadable
NFS export check : exports should not be exported globaly
NFS import check : NFS mounts should have the "nosuid" option set
promisc check : checks if network cards are in promiscious mode
list modules : just lists loaded modules
list sockets : just lists open ports

The following weekly checks are done:
password check : runs john to crack the password file, user will get an email notice to change his password asap
rpm md5 check : checks for changed files via rpm's md5 checksum feature
suid/sgid check : lists all suid and sgid files
exec group write : lists all executables which are group/world writeable
writable check : lists all files which are world writable (incl. above)
device check : lists all devices

Along these lines another system check tool should be used called “chkrootkit”. It should be run regularly and frequently at the administrator’s discretion. The chkrootkit tool is used for detection and isolation techniques. In summary it looks for the following, taken from http://www.chkrootkit.org:

chkrootkit: shell script that checks system binaries for rootkit modification. The following tests are made:
aliens asp bindshell lkm rexedcs sniffer wted scalper slapper z2 amd basename biff chfn chsh cron date du dirname echo egrep env find fingerd gpm grep hdparm su ifconfig inetd inetdconf init identd killall ldsopreload login ls lsof mail mingetty netstat named passwd pidof pop2 pop3 ps pstree rpcinfo rlogind rshd slogin sendmail sshd syslogd tar tcpd tcpdump top telnetd timed traceroute w write
ifpromisc.c: checks if the interface is in promiscuous mode.
chklastlog.c: checks for lastlog deletions.
chkwtmp.c: checks for wtmp deletions.
check_wtmpx.c: checks for wtmpx deletions. (Solaris only)
chkproc.c: checks for signs of LKM trojans.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

chkdirs.c: checks for signs of LKM trojans.
strings.c: quick and dirty strings replacement.

Another useful utility to be employed on this firewall is a tool not formerly supported anymore but still available called portsentry. Another tool in the IDS category, Portsentry can respond in real time to suspicious events. At least with this tool you can setup rules to defend against IP numbers that are doing something you know they shouldn't.

The following tools described next are used to help identify potential intruders or offenders. They are Perro, FWLogWatch, Funlog, Iplog, and SNARE. SNARE is a more advanced tool that is under the host based intrusion detection category but is easily configured and can be used to spot malicious activity on a system. In the current scenario Perro and Iplog, and SNARE are set to automatically start on system startup. Perro gathers TCP, UDP, and ICMP data into corresponding log files of the same name such as tcp.log, udp.log and icmp.log. The following is sample output from Perro:

```
10:16:35 05/03/03 192.168.10.15        3128   ndl-aas
10:16:36 05/03/03 192.168.10.15        3128   ndl-aas
10:17:38 05/03/03 192.168.10.15        3128   ndl-aas
10:19:21 05/03/03 192.168.10.10        3128   ndl-aas
10:19:21 05/03/03 192.168.10.10        3128   ndl-aas
10:19:21 05/03/03 192.168.10.10        3128   ndl-aas
```

The above data is from the tcp.log file from Perro and reflects Proxy activity originating from the internal network.

```
08:26:04 05/03/03 66.218.71.198        Echo Reply    0/0
08:26:05 05/03/03 66.218.71.198        Echo Reply    0/0
08:26:09 05/03/03 10.168.10.10         Echo Reply    0/0
08:26:10 05/03/03 10.168.10.10         Echo Reply    0/0
08:26:16 05/03/03 10.168.10.15         Echo Reply    0/0
08:26:18 05/03/03 10.168.10.15         Echo Reply    0/0
08:26:18 05/03/03 10.168.10.15         Echo Reply    0/0
08:26:30 05/03/03 10.168.10.20         Echo Reply    0/0
08:26:31 05/03/03 10.168.10.20         Echo Reply    0/0
08:37:35 05/03/03 66.218.71.198        Echo Reply    0/0
```

The above data is from the icmp.log file from Perro and reflects ICMP traffic from both internal and external networks.

The following data is output from saved iplog files:
May 3 04:08:45 TCP: port scan detected [ports 1180,1182] from unknown-48.agava.net [ports 2765,4896,2952,4924]

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

May 3 04:09:58 TCP: port scan mode expired for unknown-48.agava.net - received a total of 21 packets (760 bytes).
May 3 04:10:06 ICMP: (from A3-0-0-1712.DSL-RTR2.NY325.verizon-gni.net) pool-68-161-214-165.ny325.east.verizon.net: port is unreachable to (tcp: dest port 1552, source port 2814)
May 3 04:10:06 UDP: dgram to optima-vnet from ns7.attbi.com:53 (510 data bytes)
May 3 04:10:06 UDP: dgram to optima-vnet from ns7.attbi.com:53 (209 data bytes)

As you can see this log is more descriptive and indicative of potentially nefarious activity.

The tools Perro, Iplog, and SNARE are set-up using logrotate to provide concise logs on a daily log rotation.

FWLogWatch is a tool that can be utilized in various ways but is used in this setup to provide easily an readable summary for quick reference in HTML format that reflects the busiest IP's.

Funlog is a simple tool that is used in real time for monitoring purpose to provide a quick alert for anything that is suspicious.
The following is sample output from Funlog:

```
UDP  packet      from     192.168.1.1 on ( 1900)
UDP  packet      from     192.168.1.1 on ( 1900)
ICMP echo request from   216.218.181.2
UDP  packet      from  204.127.198.19 on ( 1053)
ICMP echo request from   216.218.181.2
ICMP echo request from   216.218.181.2
```

Funlog shows in brevity, any packet activity on any interface specified during startup. In this instance it is running on the external interface and shows 2 things. The annoying UDP packets from a Windows machine doing UpnP discoveries perhaps and ICMP requests from a foreign host that are dropped because our SuSEfirewall2 script is set to not allow ping's to the firewall.

These tools mentioned above can be combined to gather useful forensic information for use in detection, localization, identification , and assessment.

SNARE is an excellent accompaniment to Perro, Iplog, Funlog, FWLogWatch in assessing and identifying suspicious activity.

It is important in this scenario to ensure the time server is configured correctly and the log rotation schedule using log rotate is feasible. An excellent resource is

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

http://www.ntp.org/index.html where an appropriate public NTP timeserver can be selected and the client can be downloaded.

Vigilant monitoring of logs and keeping up to date with each open source tool's evolution as well as awareness of the latest exploits should enable the system administrator of such a setup as described above to deliver security and functionality in most environments using Linux.

MANAGEMENT

There are a couple scenarios to consider when deciding how to maintain the firewall and manage the network. Typically a strong firewall setup using Linux for example should offer no services. But say in our scenario there is no internal server such as a domain server that client's use. And we need to access the firewall to review the activity. Most of the time an advanced firewall is setup, there is a server on the private network in which most of the admin time is spent, and not on the firewall. Since our setup has no internal server to bounce from and we need access to the firewall we'll have to enable SSH.

If an internal server was utilized we could setup SSH in a way that a cracker would have to break into the internal system and initiate everything through the reverse-nat port (S) provided by our firewall. This would add a level of complexity that would server as a good deterrent for most crackers.

An example setup using a PIX515e firewall requires RADIUS authentication from a specific internal server using an active directory domain. All access from the public network to the firewall is blocked. In order to access the firewall we would need to establish a VPN session, which uses 3DES and MD5 and authenticates to a Windows server running Funk software's steel-belted radius server that verifies the username and password to a specific domain server running active directory. Then after obtaining access to the internal network we could telnet to the PIX firewall.

In our Linux example we have a much more simplified setup, however SSH can be used in a couple different ways and this specific setup requires that we access the firewall since all logging occurs in it. Again it is recommended that only main console access be allowed with no other system users added. But, limiting access to only SSH and under special circumstances provides substantial security and functionality to the firewall.

SSH can be configured to authenticate using keys instead of passwords to discourage brute force attempts. Moreover these keys can be allowed only from certain hosts. Since SSH is properly maintained and updated regularly and is the remote maintenance tool of choice for most network administrators it should be sufficient for the purposes of the network so far described in this paper.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

There are several resources available that describe port forwarding and tunneling with SSH if a more advanced configuration is necessary.

Assuming the Linux firewall has been setup and functioning for some time we can review the different log files. A typical function of the network administrator is to review various logs on a regular basis. A decent schedule using the outline above and the tools employed in the firewall would be a daily review in the morning as part of the network administrators first hour or two of business for the day. This time should be spent reviewing all of the necessary logs. Each log should be opened up and reviewed. Depending on activity it can be quite cumbersome to continuously look line by line at all the logs and perusing each line may be too much. But, browsing the logs looking for some obscure activity is a minimum requirement. This forces the network administrator to learn about his network and this method can assist in fully understanding different network patterns and functions. In our setup, all the logging activity is recorded in the /var/log directory. We are concerned with the four directories in /var/log among others. Primarily we would review the audit, iplog, perro, and nabou directories. Furthermore, the /var/log/firewall and /var/log/messages files contain all system messages as well and although there is some redundancy this setup provides many sources of information. Inside each of those directories are the corresponding logfiles, which are all rotated and archived nightly. There is also the current running log file that can be opened and reviewed if desired. Initially upon review the administrator can see the log file sizes. If increased traffic was expected then it should be easily discernible by size. The previous day activity has been of course zipped so the administrator could unpack and view the appropriate log file using ‘less’. The administrator should be able to eventually identify the type of traffic and the purpose for it by looking at these logs. The network administrator should be knowledgeable about packets, ports and network protocols. I recommend initially reviewing the Portsentry logs. Depending on how they were setup the administrator can easily see if any host was blocked by the naming convention portsentry.blocked.[protocol]. Protocol would be tcp or udp for example. If nothing is detected after consulting the Portsentry log the network administrator should review the Nabou report. This report can be used to quickly identify any file changes and system modifications and is quite comprehensive and easily configurable. After reviewing Nabou and depending on how the chkrootkit was configured, if applicable run or review the chkrootkit output.

Obviously the network administrator should become aware of the current system status when accessing the system for the first time of the day. For example the administrator should be aware of the current time and correlate the logs accordingly to see if anything is out of the ordinary and while reviewing the logs looks for any inconsistencies. The iplog and perro logs should be opened up and quickly browsed looking for repetitious activity, which can be normal such as consistent activity from the internal hosts using the proxy service. Or identify activity that appears to be “reconnaissance like” using multiple protocols. ICMP

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

for example is mostly used for troubleshooting or diagnostics and can be frequently used by legitimate applications but is also easily identified when used for malicious intent.

Another nice tool for reviewing some of the default system logs is called logtool. This tool can present the data from those logs in a more pleasing display that is easier to read. This tool by default can be used to read the messages and firewall logs. The author of logtool states the main uses of it:
“As an engine for use by shell scripts to generate nice little reports to e-mail people. As a tool to use in conjunction with simple shell scripts for automated webpage generation. As a logfile monitoring tool for network operations center like environments, where it's nice to have logfile's scrolling across the screen in easy to read color. I've included a "redbeep" option so that in such cases, "red" events will generate a Ctrl-G to catch the attention of NOC personnel.”

Fwlogwatch is another configurable tool that has some other abilities but its strongest use is to quickly parse the log and display activity that has crossed predefined thresholds.

The audit log recorded by SNARE is the more complex log and should be treated accordingly by spending most of the admin time in it. The network administrator should be familiar with normal maintenance activity that is regularly performed and distinguish this with other activity. Anything suspicious can be easily correlated with other logs should such activity arise. There is a graphical front-end to view SNARE output but assumes you have a desktop such as gnome. Since our setup does not include X windows we can browse the output using the command line.

Another useful tool that should be used at the network administrator’s discretion is Fcheck. This tool is a simple system integrity checker without any cryptographic ability. It’s extremely fast and requires nominal system resources. Since we are using a more complete tool in Nabou that runs nightly, the purpose of Fcheck is to perform random checks or simply run checks more frequently. Although it lacks in security features it “is fast, free, highly portable, and simple, but also makes some notable tradeoffs at security’s expense.” (Bauer, p.383) In our scenario the network administrator may not check on the firewall but once per day. Running Fcheck may assist in quickly spotting system inconsistencies should any occur. This could save time if a system is breached since we can schedule Fcheck to run at regular intervals in a 24-hour period and use the system logger instead of standard output for later review. The trade off in security using Fcheck is more plausible here since our system is setup to thwart external hacks and our firewall of course has no regular system users. And that it is extremely fast is a good supplement to the Nabou checker that augments overall security.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In conclusion the firewall setup described above is not overly complex yet highly functional and simple enough for regular management. Although an intermediate level of skill is required to set the system up, once running, a policy and procedure manual can be easily composed to form a logical set of system checks ensuring that reasonable care has been taken to provide security. Furthermore, we have system that can be supported "with highly trained people who can do the assessment, patches and alarm correlation, decipher the logs and actually deter intrusions."

"We need good boxes automating what they can…but it's only with good people using the tools that you get the best outcomes." (Cummings)

Reference:

Bauer, Michael. "Building secure servers with Linux". O'Reilly. 2003

Krishna, Arvind. "Five Steps for keeping Hackers at bay." 18 Sep 2002.
URL: http://zdnet.com.com/2100-1107-958397.html

"How to develop a Network Security Policy."
URL: http://wwws.sun.com/software/whitepapers/wp-security-devsecpolicy/

Mackey, Richard. "Security Architecture." Jun 2002.
URL: http://www.infosecuritymag.com/2002/jun/insecurity.shtml

Annis, William. "The Death of Monitoring."
URL: http://www.biostat.wisc.edu/~annis/mom3/

Cummings, Joanne. "The people side of prevention." Network

World. URL: http://www.nwfusion.com/buzz/2002/intruderside.html (Sept. 23, 2002).

Schneier, Bruce. Crypto-Gram Newsletter. May 2000

URL: http://www.counterpane.com/crypto-gram-0005.html#ComputerSecurityWillWeEverLearn

Fwlogwatch
URL: http://www.kyb.uni-stuttgart.de/boris/software.shtml

Nabou
URL: http://www.co.daemon.de/en/

Harden Suse
URL: http://www.suse.com/~marc/harden_suse.html

SuSE Security Checker

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

URL: http://www.suse.com/~marc/seccheck.html

Chkrootkit
URL: http://www.chkrootkit.org/

SNARE
URL: http://www.intersectalliance.com

SuSEfirewall2
URL: http://www.suse.com/~marc/SuSEfirewall2.html

IPLog
URL: http://dev.ojnk.net/

Perro
URL: http://www.grigna.com/diego/linux/perro/

© SANS Institute 2003, Author retains full rights

Last Updated: June 19th, 2013

Upcoming SANS Training

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

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**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**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 Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Melbourne 2013	**Melbourne, AU**	**Sep 02, 2013 - Sep 07, 2013**	**Live Event**
SANS Capital City 2013	**Washington, DCUS**	**Sep 03, 2013 - Sep 08, 2013**	**Live Event**
Network Security 2013	**Las Vegas, NVUS**	**Sep 14, 2013 - Sep 23, 2013**	**Live Event**
SANS Canberra 2013	**OnlineAU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**