Global Information Assurance Certification Paper

Foo.com
Security Audit
July 25, 2000

Executive Summary:

The overall rating at Foo.com is very poor. Immediate action is required. The total lack of security measures guarantees any hacker attempt, to be successful. There are no company security policies; minimal systems based security, and non-existent network intrusion detection.

The company currently has no policy regarding system security. I created a "banner" based login warning that informed users of the usage guidelines. I also suggested a set of parameters on which they could base a company-wide systems usage policy. The costs for these changes are minimal to none.

For new machines, some of the suggestions I made was to install the machines on a "dead" network and install the smallest software cluster and add packages afterwards. I also made some reminders to install patches, which are vitally important.

The next step was to secure the operating system on installed machines. Removal of boot files, locking down of system ports, and some network interface configurations are covered.

Drive / file system configurations were next in the checklist. I discussed the mounting and file system options that will help secure the systems.

System based application are another very large security hole in Solaris systems. Programs such as rpc, nfs, and nis are very dangerous. I made several suggestions on which I could proceed in fixing these holes. Some ways are to close non-used ports and / or use programs to limit access or block status requests.

Third party applications were another issues that I covered. The primary source of concern was from the Apache / Stronghold webserver. The program was compiled with minimal security. The internal server options were also set without any security in mind. The Java and load balancing programs open a very large number of ports that need to be watched. The WebMethods server utilizes it's own web server that has absolutely no security; it's best to disable this function.

Other tools are needed to be installed to maintain a secure environment. Programs that monitor the log files are very useful. They allow the system administrator to work on other stuff while it sorts through the log files. Programs that monitor file integrity are also important tools to track down unscheduled system changes. Upgrading the system-logged daemon is also a good idea.

Firewalls and intrusion detection are core tools in network security. I recommended a fail-over redundant firewall system. I also suggested the purchase of an internal network detection utility system. This will watch the internal LAN for unusual traffic.

Some other items I discussed were user password creation, a secure log server, running a time synchronization system, and protecting DNS and sendmail servers.

Company policies:

After speaking with the Chief Technology Officer and the Director of Human relations, I discovered that there was currently no policy for maintaining a reasonable level of systems security. The company's systems did not have any type of authorization message upon system login nor at any other usage time. In addition there was no procedure to ensure that company personnel knew what and what not to do on the systems. E-mail guidelines and usage of network intensive applications were also left to each user's whims.

To remedy these issues I recommended the following changes:

1) The addition of an authorization message to all company systems. An example I gave them is:

> This system is for the use of authorized users only. If you are unauthorized, you are required to logout immediately. Individuals using this computer system without authority or in excess of their authority are subject to having all of their activities on this system monitored and recorded by system personnel.
>
> In the course of monitoring individuals improperly using this system, or in the course of system maintenance, the activities of authorized users may also be monitored.
>
> Anyone using this system expressly consents to such monitoring and is advised that if such monitoring reveals possible evidence of criminal activity, or any activity that violates the Foo Systems Agreement Form, system personnel may provide the evidence of such monitoring to law enforcement officials.

2) A creation of a Systems Agreement form, using the following guidelines[1]:

    1) E-mail
        a) Is e-mail personal or company owned?
        b) Spamming Rules (Internal and External)
        c) Sending Large attachments

    2) Network Applications
        a) Are there a limited number of applications the users may run?
        b) Rules for non-business network bandwidth hogs. I.E. Napster, IRC, streaming media, etc.
        c) Procedures for running security / hacker type scanners / tools

    3) Logins
        a) Individual passwords must be kept secret

[1] Complete layout of form located in Section 2

b) Root / Admin passwords are only given to Administrators
c) Password lists are only given to Administrators
d) Login time restrictions
e) Machine restrictions
f) Usage of one-time-passwords / encrypted login sessions

4) Physical Contact
    a) Those without authorization may not voluntarily come in contact with any systems equipment.
    b) Any accidental contact should be reported immediately to an administrator.

5) System Usage
    a) Persons may use what they need to complete their work and should not "experiment" with other system resources
    b) Users should understand what their authorization level grants them access to and should not exceed that authority.
    c) No one should be installing any non-company services / programs on company servers. E.g. entering "my.domain.com" in the company's DNS server.

The costs of the changes are minimal to none. The only cost incurred is the time and effort put in to creating the documents, "legal" looking it over, and having the staff sign the agreement.

System: Operating System: Installation

To ensure the most protection for your systems, several steps need to be taken. In your current configurations it appears that the systems were not optimally configured. Upon speaking with the systems administrator, I have found that the systems were installed on a live network and the full OEM software package was installed. These pose some concern.

Installing a machine on a live network allows hackers to target a system that has not yet been prepared for security attacks. None of the protection or logging has yet been installed, leaving the machine wide open. There are several instances of systems being hacked only 5 minutes into its connection to a live network. Secure machines should always be installed on "no network" or an isolated network, for jumpstart situations. Physical security is also important during an install, find a locked room to do an install.

Installing the full OEM software package is also not recommended. The more that gets installed, the more a hacker has to work with. Systems should only be installed with what you need it to run. It's much easier to add components to a machine than it is to take clusters away at a time. While the "core" cluster would have been appropriate for you webservers, your webservers are also your application servers. Because of this, I would recommend the developer cluster, due to the libraries are installed at this level. The use of "X" also points to this. If you have the time though, installing the "core" cluster and adding, "X" and the libraries / include packages is more secure. Other packages that will be useful for environment are (all with from Sun and therefore the SUNW header applies): nptr and ntpu (for ntp, discussed later), libm and libms (for Perl), ter (for terminal information), accr and accu (for system accounting), and scpu (for BSD commands).

Patches are another important step in securing a system. You patches seem to be slightly out of date. Patches are important because many bug and security fixes are distributed in patches. This has more recently become even more important due to the rumor that Solaris is going to become open source. This means hackers will have access the code base and therefore more intimate knowledge of how the code works and how to hack it. Sun's patch report should be checked often for updates.

System: Operating System: Configuration

The first step is the system boot files. The boot files reside in /etc/rc#.d. These directories need to be looked over and the services that are not going to run, need to be disabled. I am not a fan of deleting configuration files; in the rc directories you can simply rename the start scripts (change the S to a s). This will allow you to reactivate the service if you need to. I also don't like to remove any kill scripts; it is preferable to have a try to kill a non-existent process rather than letting it die as the system goes down. The most important rc directories to look at is the rc2.d, rcS.d, and rc3.d, the rest are primarily kill scripts.

```
/etc/rc2.d:
S01MOUNTFSYS
S05RMTMPFILES
S20sysetup
S21perf
S30sysid.net
S40llc2
S47asppp
S69inet
S70uucp
S71rpc
S71sysid.sys
S72autoinstall
S72inetsvc
S73cachefs.daemon
S73nfs.client
S74autofs
S74syslog
S74xntpd
S75cron
S75flashprom
S75savecore
S76nscd
S80PRESERVE
S80lp
S80spc
S85power
S88utmpd
S89bdconfig
S91afbinit
S92volmgt
S93cacheos.finish
S98ssh
S99audit
S99cdagent
S99dtlogin
S99reporter-agent
S99tsquantum
S88sendmail

/etc/rc3.d:
S15nfs.server
S76snmpdx
S77dmi
S99foo

/etc/rcS.d:
S10cvc
S10initpcmcia
S30rootusr.sh
S33keymap.sh
S35cacheos.sh
S40standardmounts.sh
S41cachefs.root
S42coreadm
S50devfsadm
S70buildmnttab.sh
```

The first configurations that should be disabled are the ones pertaining to nfs. Nfs contains many security holes and since your environment does not require it, it should be disabled. The "core-file" based scripts should be next. While the core file does have some beneficial uses, on a production machine it's more trouble than it's worth. On machines with a lot of ram a hacker-forced core dump can act like a denial-of-service attack, also a hacker can find restricted information in a core file that can be used to attack the system in other ways. There are more steps required to stop core dumps, which I will talk about a little later. Various other services should be disabled due to lack of use and security holes. The power scripts are some because your machines are servers and are not running on battery power. In addition these machines do not have any need to print directly so the lp service can be deactivated. The rpc-based scripts are also not needed in your architecture. Snmp is also not implemented and therefore disabled.

Some other security steps need to be taken. Setting permissions for system daemons and numerous changes to the network interfaces need to be done. The daemon

permissions can be set with a minor script in the /etc/init.d directory. The network parameters need to be set in the /etc/init.d/inetinit file. There are many configuration steps needed there. These changes help stop many network based hacker attacks, such as "SYN-flooding" and "ARP-spoofing attacks." Also disabling Solaris' feature allowing it to be a router is important. Your environment allows for static routing, which is the most secure and so, will be used. Lastly an alteration of the /etc/init.d/inetsvc file, to disable DHCP, multicast routing, and inetd services (inetd is talked about in the next section).

System: Operating System: Drive / Directory / File System Configuration

This is a copy of your /etc/vfstab file, which contains your drive configurations.

```
#device        device         mount          FS      fsck    mount   mount
#to mount      to fsck        point          type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr              ufs     1       yes     -
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t0d0s3       -       -       swap    -       no      -
/dev/dsk/c0t1d0s0       -       -       swap    -       no      -
/dev/dsk/c0t0d0s0       /dev/rdsk/c0t0d0s0      /       ufs     1       no      -
/dev/dsk/c0t0d0s6       /dev/rdsk/c0t0d0s6      /usr    ufs     1       no      -
/dev/dsk/c0t0d0s1       /dev/rdsk/c0t0d0s1      /var    ufs     1       no      -
/dev/dsk/c0t1d0s1       /dev/rdsk/c0t1d0s1      /home   ufs     2       yes     -
/dev/dsk/c0t0d0s5       /dev/rdsk/c0t0d0s5      /opt    ufs     2       yes     -
/dev/dsk/c0t0d0s7       /dev/rdsk/c0t0d0s7      /tmp    ufs     2       yes     -
/dev/dsk/c0t1d0s3       /dev/rdsk/c0t1d0s3      /web    ufs     2       yes     -
```

Multiple partitions are very important on all systems. It allows the management of file-systems and stops some hacker attacks. A core dump attack can be averted, in part, by correctly configured partitions. Generation of core files can be stopped altogether by a minor alteration to the /etc/system file (add: set sys:coredumpsize=0).

There are some modifications needed in the drive configuration. Some options can be set to further secure the system. The nosuid options tells the system to ignore any files that have a setuid permission and to ro options tells the system that the directory is to be set read-only. Setuid files are dangerous at all levels but they can't all be removed. Some are required for system operations. In addition the nosuid option also means nodev or no devices. That means that the nosuid option cannot be used on and system directory containing devices But there are some directories that can be set nosuid. In your case, /var, /web, and /home can all be set nosuid. This will prevent any setuid files from being created or copied in / to those directories. The /usr directory can be set ro to protect the system binaries and libraries.

Within file systems there are some issues too. A very common hacker attack is the now infamous "buffer overflow" attack. This causes a program to crash and give the hacker a login shell with the permissions of the person who ran the crashed program (usually root). The fix is quick and painless, all it requires is another minor alteration to the /etc/system file (add: set noexec_user_stack=1). World-writable directories are another problem. These can be fixed quickly by the use of a program called "fix-modes" by Casper Dik. Fix-modes corrects permissions on files and directories by removing group and world write permissions and checks that most files belong to root, that should. Yassp and Titan are two other programs that help to lock down a system.

System: Operating System: Applications

There are many vulnerabilities in many system defaults. Many system settings and default applications are prime targets for intrusions. Approximately 90% of the default internet services are security holes, either allowing a hacker to gain account access or creating a denial of service attack against your system or using your machine to attack another system. Your internet services are controlled by the /etc/inetd.conf file.

```
ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd      in.ftpd
#telnet stream  tcp     nowait  root    /usr/sbin/in.telnetd
name    dgram   udp     wait    root    /usr/sbin/in.tnamed    in.tnamed
shell   stream  tcp     nowait  root    /usr/sbin/in.rshd      in.rshd
login   stream  tcp     nowait  root    /usr/sbin/in.rlogind   in.rlogind
exec    stream  tcp     nowait  root    /usr/sbin/in.rexecd    in.rexecd
comsat  dgram   udp     wait    root    /usr/sbin/in.comsat    in.comsat
talk    dgram   udp     wait    root    /usr/sbin/in.talkd     in.talkd
uucp    stream  tcp     nowait  root    /usr/sbin/in.uucpd     in.uucpd
tftp    dgram   udp     wait    root    /usr/sbin/in.tftpd     in.tftpd -s /tftpboot
finger  stream  tcp     nowait  nobody  /usr/sbin/in.fingerd
#systat stream  tcp     nowait  root    /usr/bin/ps            ps -ef
#netstat        stream  tcp     nowait  root    /usr/bin/netstat
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
echo    stream  tcp     nowait  root    internal
echo    dgram   udp     wait    root    internal
discard stream  tcp     nowait  root    internal
discard dgram   udp     wait    root    internal
daytime stream  tcp     nowait  root    internal
daytime dgram   udp     wait    root    internal
chargen stream  tcp     nowait  root    internal
chargen dgram   udp     wait    root    internal
100232/10       tli     rpc/udp wait root /usr/sbin/sadmind   sadmind
rquotad/1       tli     rpc/datagram_v wait root /usr/lib/nfs/rquotad        rquotad
rusersd/2-3     tli     rpc/datagram_v,circuit_v      wait root
sprayd/1        tli     rpc/datagram_v wait root /usr/lib/netsvc/spray/rpc.sprayd
walld/1         tli     rpc/datagram_v wait root
rstatd/2-4      tli   rpc/datagram_v wait root /usr/lib/netsvc/rstat/rpc.rstatd
#rexd/1           tli  rpc/tcp wait root /usr/sbin/rpc.rexd     rpc.rexd
100083/1        tli     rpc/tcp wait root /usr/dt/bin/rpc.ttdbserverd
#ufsd/1 tli     rpc/*   wait    root    /usr/lib/fs/ufs/ufsd   ufsd -p
100221/1        tli     rpc/tcp wait root /usr/openwin/bin/kcms_server
fs              stream  tcp     wait nobody /usr/openwin/lib/fs.auto  fs
100235/1 tli rpc/tcp wait root /usr/lib/fs/cachefs/cachefsd cachefsd
kerbd/4         tli     rpc/ticlts      wait    root    /usr/sbin/kerbd     kerbd
printer         stream  tcp     nowait  root    /usr/lib/print/in.lpd
100234/1        tli     rpc/ticotsord  wait    root    /usr/lib/gss/gssd gssd
dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd
100068/2-5 dgram rpc/udp wait root /usr/dt/bin/rpc.cmsd rpc.cmsd
300326/4        tli     rpc/tcp wait    root    /platform/SUNW,Ultra-Enterprise-
10000/lib/dr_daemon    dr_daemon
```

After reviewing the active configuration it appears that the default settings are in use. Here are the primary concerns for your machines. Remember, that anytime your machine is hacked it can be used as a weapon against other systems:

FTP File transfer protocol: The service allows users to conduct file transfers to and from a system. While this service is active, a security hole exists that allows hackers a way into the system. The two most dangerous problems are; 1) a hacker can navigate your file-system and retrieve files, as well as store harmful or illegal files on your system.

2) Using a buffer overflow attack a hacker can gain access to a user account, sometimes the super-user's account.

Telnet is a service that allows a command-line based connection between two systems. This service is a large concern because it allows a hacker to have shell access to your system. There are several ways a hacker can gain access through telnet; because account information is sent unencrypted a hacker can use a sniffer to collect account names and passwords. Buffer overflow and brute-force attacks can also be used.

TFTP Trivial File Transfer Protocol: Is another service that allows the transfers of files. This service is even more dangerous than FTP because it doesn't require a login. It allows any access by default. It has the same security problems that FTP has, but with not even the login for security it is much easier to exploit this service.

Finger Finger is a service that allows local and remote users to get information about other users on a system. This has an obvious and not so obvious problem, giving out any information is usually a bad thing. The less a hacker knows about anything on your system the better. But, in addition to giving some minor information, sometimes the finger daemon gives more information than it should.

R commands: Login, exec, and shell are all commands that allow users to have access to command prompts. These are more clear-text based connection services, similar to telnet. Their exploits are also similar to telnet's.

REXD Remote Execution Server: This allows users to execute programs remotely. This is another blatant security hole. Hackers can simply brute-force or sniff the command information and exploit it and run programs off your system.

Other The other services run by the inet daemon are also dangerous. They all allow a hacker to: 1) get information about the system, 2) run programs on your machine, 3) execute a denial of service attack against your systems or cause your machines to attack another system. For example "echo" and "chargen" are two services that run by default. Echo mimics any input, while chargen generates a random string of characters. A hacker can point the two services at each other and cause the system to fill the network with useless random characters.

The best way to protect your systems from these security risks is to disable all of them. Most of the services that run by default are not used anyway. For the few systems in which you must run there are several ways to add some security to them.

The best utility to use with most internet services is Tcp-wrappers by Wietse Venema. The tcp-wrappers program allows the administrator to control which IPs are allowed to access a specific service on your systems. It can also be set to take some actions based on login attempts. The wrapper can be configured to log and connection and try to get some additional information about the attempt by using "finger."

There is another problem that occurs when these services run. Each service opens a port on your system. These ports can be attacked directly or they can be used to gain

information about your systems. One program that can help protect your systems ports are rpcbind and portmap also by Wietse Venema. These programs offer the same type of security additions as tcp-wrappers. Rpcbind fixes several rpc bugs as well as allowing IP filtering for security. Portmap actively monitors your system's ports and does IP filtering; it also dumps bad connections, not revealing the status of the port to a hacker.

The best resolution for the lack of security in telnet and ftp is to disable telnet and ftp and use Ssh. Ssh replaces and functionality of telnet, ftp, rcp, rsh, and rlogin. It also allows other protocols to be sent though it (tunneling). Ssh provides several different types of encryption as well as compression features. This encryption makes it very difficult for hackers to intercept and use any information that passes between your systems.

The COPS and nmap reports (Section 2) were run and the finding validating my suggestions.

Systems: Third Party Applications

As shown in the following output…

```
Ares :/etc/rc3.d #>ps -ef|awk '{print $1 "...." $8 "    " $9 $10 $11}'
UID....CMD

{lines deleted}

1.    root....0:07    /usr/java1.2/bin/../jre/bin/../bin/sparc/native_threads/javacom.livesoftware.j
2.    root....0:27    /usr/bin/../java/bin/../bin/sparc/native_threads/oldjavadbConnectionUtils.jdbc
3.    root....0:03    /bin/../java/bin/../jre/bin/../bin/sparc/native_threads/javacom.livesoftware.j
4.    root....5:47    /usr/java1.1/bin/sparc/native_threads/java-ms64M-mx64M
5.    root....0:06    /usr/bin/../java/bin/../jre/bin/../bin/sparc/native_threads/rmiregistry
6.    www....0:00    ./httpsd-d/opt/stronghold
7.    root....0:00    ./httpsd-d/opt/stronghold
8.    www...../httpsd    -d/opt/stronghold-f
9.    root....0:00    ./httpsd-d/opt/stronghold
10.   www....0:00    ./httpsd-d/opt/stronghold
11.   root....0:00    /bin/sh/var/resonate/bin/reporter-agent
12.   root....0:00    /var/resonate/bin/cdagent
13.   root....0:00    /var/resonate/bin/reporter-agent.bin-f-p
14.   root....297:47    /var/resonate/bin/cdagent
15.   root....0:00    /bin/sh/opt/webMethods/Server/bin/server.sh
16.   root....0:00    /usr/lib/sendmail-bd-q15m
17.   root....0:00    /usr/local/sbin/sshd2

{lines deleted}

Ares :/etc/rc3.d #>
```

There are several applications on the system that are third party. Many programs exist that can collect an enormous amount of data about your systems without you even knowing it. Some of which I have used to test your current security status. Nessus, SARA, and nmap are some of them. While nmap just scans systems for open ports using different types of scan procedures, the other tools are far more extensive. Section 2 contains all the output of the tools I used to do the penetration tests.

Using Nessus, which in my opinion is the best security scanner, I was able to retrieve a considerable amount of data about your machines. Nessus generated a list of all your system's open ports, what services were running on most, the operating system that you were running, and which port was the most dangerous, security-wise.

Your web server is what we'll discuss first. You are running Stronghold v 2.4.2. Stronghold is an SSL capable version overlying Apache (v 1.3.8 in this case). My first suggestion is to upgrade to Stronghold v3, which contains Apache 1.3.12. This upgrade has several bug and security fixes. Some changes I would recommend for the new compile is to add the mod_auth_digest module, which requires the installation of the /dev/random library. This module will allow MD5 encryption of password / username information. In the httpd.conf file we're making sure that secure directories require logins, also that harmful options are not allowed (FollowSymLinks, ExecCGI, Indexes, AllowOverride, etc.). Setting the default access to deny is the next step. This will protect certain directories from unwanted users. You have already setup and run SSL using a Verisign certificate, this is good. Lastly, if you so choose, locked the whole directory structure with chroot is another good idea.

This box is also running sendmail but, it is not a mail server. It is not recommended to run sendmail in this configuration due to sendmail's security problems. Since this box is not a mail server it is best just to have the sendmail program run once and a while to make sure the mail queue is clean. Sendmail does not have to run in order for the box to be a mail client.

There are many other products running on this box. It appears that several instances of Java are running. These Java programs have opened many external system ports (59xxx), as shown in the Nessus report and the lsof utility (lsof data unavailable at this time). The WebMethods program also opens an external port but, this one is particularly dangerous because it uses a proprietary webserver process for configurations. This can be hacked and do considerable damage t your services. Lastly there are a large number of ports opened by the resonate (A software based load-balancer) processes. After studying your architecture it appears that these services and port must remain open to function correctly. The only system-based change that can be made is to disable the WebMethods webserver and do all configurations from the command line. The other issues will have to be addressed by another implementation, a firewall (see Network Section, below).

System: Maintaining Order

Even if a system has been locked down and secured, it must still be watched carefully. There are many programs that will help your administrator do this. The first of which is Tripwire by Gene Kim and Gene Spafford.

Tripwire is a file integrity program. It verifies the stability of files based against a "snapshot" taken when the system was in a clean state. The configuration file allows you to determine which directories and files you want the tripwire program to watch, as well as what parameters to watch for alterations in. It is important to keep the tripwire database on non-writable media. My suggestion is to put it on a CDRW disk. The disk can be written to only if the administrator is at the system and physically removes the disk and puts it in a burner. Your system, being in a data center ensures that no one else will have access to your machines.

Another useful too that I highly recommend is syslog-ng from Balazs Scheidler. Syslog-ng as the name shows is a syslogd replacement, but with new functionality. The original syslogd allows messages only to be sorted based on priority/facility pair, syslog-ng adds the possibility to filter based on message contents using regular expressions. The new configuration layout is very powerful. This makes it much easier to go through the log files and to comprehend the messages.

Log parsing is another task that can be automated. I prefer Swatch to Logcheck, which are the two most popular log parsers. Swatch offers a Perl-type language, which is useful and the action response is very useful. Swatch can mail or write to terminal so the mail cannot be intercepted, a feature lacking in Logcheck. Although it does require a higher level of log understanding in the administrator.

LSOF is a "must have" tool. It has too many uses to list but the primary uses I have found are; listing the processes using a specific port and listing the libraries that a process is calling, and unlinked file listing.

Network: Firewall / Intrusion Detection

A firewall is an integral part of any company's security setup. A firewall can block many hacker attempts to subvert your systems. Based on your architecture and requirement, I recommend a dual CheckPoint Firewall-1 setup running on two locked-down Sun Ultra 10 systems and using the Stonebeat software for failover on the firewall systems.

Your layout has a special requirement that demands a complete firewall system. The large numbers of processes that activate external ports on your boxes that only require LAN-type access. To compensate for this the firewall system must block all of those external ports from leaving the LAN environment. In your environment only the following ports need to be accessed by the Internet:

80:	Web Serving
443:	Web Serving, SSL
569:	SSH
2101:	Resonate Controller

The rest of your services do not need to be externally accessible. The StoneBeat software will allow you some extra protection. If your primary firewall ever fails a second system will automatically be engaged. This will ensure total firewall coverage.

In addition to a firewall an internal LAN-based intrusion detection system (IDS) should be deployed. These systems will watch the LAN segments for questionable transactions. ISS's RealSecure is a very good IDS. RealSecure uses a standards-based approach, comparing network traffic and host log entries to the known and likely methods of attackers. Suspicious activities trigger administrator alarms and other configurable responses.

Miscellaneous:

Passwords are a hassle for both users and administrators. Until programs like Passwd+ become more compatible with current systems, I recommend enforcing company-wide policies about creating and changing passwords. Setting the password lifetime is another good policy. Running a password cracker like Crack, is a good idea to weed out users who use bad password schemes.

Purchasing a separate loghost machine is a very good idea. This will force hackers to break into two machines in order to cover their tracks, increasing their chance of being caught. This also lowers your need to backup all your machines just to get their log files.

NTP is a protocol that synchronizes time across your machines. This is very important when dealing with log machines and databases. The best way to implement NTP is to buy a stratum 1 server. This will allow perfect time among your internal machines.

Backups are one of the most important procedures to have. The best way to back your systems up is to add additional network interfaces to your machines and back them up over a private LAN. I recommend a moderately sized DLT library and Legato back-up software. Backup and ape rotation schedules should be planned before implementation.

Although my audit didn't include you DNS and mail servers, I have some suggestions about them. For DNS and sendmail make sure you chroot both environments, to lock down and "break-out" attempts. For DNS make sure that the allow / deny options are in effect, to stop hackers from requesting your entire DNS tables.

Nessus Report

(Files attached separately: They do not seem to port well)

The Nessus Security Scanner was used to assess the security of 1 host

- **23 security holes have been found**
- **30 security warnings have been found**
- **5 security notes have been found**

Part I : Graphical Summary :

This file was generated by Nessus, the open-sourced security scanner.

COPS Report

```
ATTENTION:
Security Report for Mon Jul 17 09:39:20 EDT 2000
from host ares


**** root.chk ****
**** dev.chk ****
**** is_able.chk ****
Warning!  /etc/security is _World_ readable!
Warning!  /usr/adm/spellhist is _World_ writable!
Warning!  /usr/adm/vold.log is _World_ writable!
**** rc.chk ****
**** cron.chk ****
**** group.chk ****
**** home.chk ****
Warning!  User nuucp's home directory /var/spool/uucppublic is mode
01777!
**** passwd.chk ****
**** user.chk ****
**** misc.chk ****
**** ftp.chk ****
Warning!  /etc/ftpusers should exist!
**** pass.chk ****
**** kuang ****
**** bug.chk ****
Warning!  /usr/lib/sendmail could have a hole/bug!  (CA-88:01)
Warning!  /usr/lib/sendmail could have a hole/bug!  (CA-90:01)
Warning!  /bin/mail could have a hole/bug!  (CA-91:01a)
```

Nmap: Syn Scan

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host ares.foo.com (100.100.100.246) appears to be up ... good.
Initiating SYN half-open stealth scan against ares.foo.com (100.100.100.246)
Adding TCP port 32773 (state open).
Adding TCP port 56492 (state open).
Adding TCP port 32771 (state open).
Adding TCP port 2101 (state open).
Adding TCP port 111 (state open).
Adding TCP port 32779 (state open).
Adding TCP port 48265 (state open).
Adding TCP port 32777 (state open).
Adding TCP port 7100 (state open).
Adding TCP port 32778 (state open).
Adding TCP port 37 (state open).
Adding TCP port 515 (state open).
Adding TCP port 55770 (state open).
Adding TCP port 513 (state open).
Adding TCP port 55768 (state open).
Adding TCP port 540 (state open).
Adding TCP port 55767 (state open).
Adding TCP port 13 (state open).
Adding TCP port 19 (state open).
Adding TCP port 8082 (state open).
Adding TCP port 512 (state open).
Adding TCP port 55769 (state open).
Adding TCP port 55766 (state open).
Adding TCP port 60018 (state open).
Adding TCP port 79 (state open).
Adding TCP port 444 (state open).
Adding TCP port 1099 (state open).
Adding TCP port 8083 (state open).
Adding TCP port 569 (state open).
Adding TCP port 21 (state open).
Adding TCP port 56494 (state open).
Adding TCP port 2161 (state open).
Adding TCP port 80 (state open).
Adding TCP port 9 (state open).
Adding TCP port 6112 (state open).
Adding TCP port 6666 (state open).
Adding TCP port 514 (state open).
Adding TCP port 25 (state open).
Adding TCP port 81 (state open).
Adding TCP port 443 (state open).
The SYN scan took 1779 seconds to scan 65535 ports.
For OS Scan assuming that port 9 is open and port 1 is closed and neither are firewalled
Interesting ports on ares.foo.com (100.100.100.246):
(The 65486 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      filtered    echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
25/tcp     open        smtp
37/tcp     open        time
79/tcp     open        finger
80/tcp     open        http
81/tcp     open        hosts2-ns
111/tcp    open        sunrpc
443/tcp    open        https
444/tcp    open        snpp
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
569/tcp    open        ms-rome
```

```
1099/tcp   open        unknown
2101/tcp   open        unknown
2161/tcp   open        unknown
6112/tcp   open        dtspc
6666/tcp   open        irc-serv
7100/tcp   open        font-service
8082/tcp   open        unknown
8083/tcp   open        unknown
13623/tcp  filtered    unknown
19319/tcp  filtered    unknown
32771/tcp  open        sometimes-rpc5
32773/tcp  open        sometimes-rpc9
32777/tcp  open        sometimes-rpc17
32778/tcp  open        sometimes-rpc19
32779/tcp  open        sometimes-rpc21
33731/tcp  filtered    unknown
38985/tcp  filtered    unknown
44580/tcp  filtered    unknown
44919/tcp  filtered    unknown
48265/tcp  open        unknown
53111/tcp  filtered    unknown
55766/tcp  open        unknown
55767/tcp  open        unknown
55768/tcp  open        unknown
55769/tcp  open        unknown
55770/tcp  open        unknown
56071/tcp  filtered    unknown
56492/tcp  open        unknown
56494/tcp  open        unknown
60018/tcp  open        unknown

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=39886 (Worthy challenge)

Sequence numbers: 454598E0 45475AC1 45491B4B 454A4DBA 454A7075 454C02C4
Remote operating system guess: Solaris 7
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=9BCE)
T1(Resp=Y%DF=Y%W=212%ACK=S++%Flags=AS%Ops=NNTME)
T2(Resp=N)
T3(Resp=N)
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=0%IPLEN=70%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)


Nmap run completed -- 1 IP address (1 host up) scanned in 1785 seconds
```

Nmap: TCP Scan

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host ares.foo.com (100.100.100.246) appears to be up ... good.
Initiating TCP connect() scan against ares.foo.com (100.100.100.246)
Adding TCP port 111 (state open).
Adding TCP port 81 (state open).
Adding TCP port 37 (state open).
Adding TCP port 55770 (state open).
Adding TCP port 2101 (state open).
Adding TCP port 79 (state open).
Adding TCP port 60018 (state open).
Adding TCP port 32779 (state open).
Adding TCP port 32773 (state open).
Adding TCP port 48265 (state open).
Adding TCP port 32771 (state open).
Adding TCP port 56492 (state open).
Adding TCP port 32778 (state open).
Adding TCP port 32777 (state open).
Adding TCP port 55767 (state open).
Adding TCP port 540 (state open).
Adding TCP port 512 (state open).
Adding TCP port 6112 (state open).
Adding TCP port 19 (state open).
Adding TCP port 56494 (state open).
Adding TCP port 569 (state open).
Adding TCP port 7100 (state open).
Adding TCP port 6666 (state open).
Adding TCP port 515 (state open).
Adding TCP port 55769 (state open).
Adding TCP port 13 (state open).
Adding TCP port 2161 (state open).
Adding TCP port 55768 (state open).
Adding TCP port 7 (state open).
Adding TCP port 55766 (state open).
Adding TCP port 4045 (state open).
Adding TCP port 514 (state open).
Adding TCP port 8082 (state open).
Adding TCP port 443 (state open).
Adding TCP port 21 (state open).
Adding TCP port 9 (state open).
Adding TCP port 80 (state open).
Adding TCP port 1099 (state open).
Adding TCP port 25 (state open).
Adding TCP port 513 (state open).
Adding TCP port 444 (state open).
Adding TCP port 8083 (state open).
The TCP connect scan took 156 seconds to scan 65535 ports.
For OS Scan assuming that port 7 is open and port 1 is closed and neither are firewalled
Interesting ports on ares.foo.com (100.100.100.246):
(The 65493 ports scanned but not shown below are in state: closed)
Port       State       Service
WARNING!  The following files exist and are readable: /usr/local/share/nmap/nmap-services
and ./nmap-services.  I am choosing /usr/local/share/nmap/nmap-services for security
reasons.  set NMAPDIR=. to give priority to files in your local directory
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
25/tcp     open        smtp
37/tcp     open        time
79/tcp     open        finger
80/tcp     open        http
81/tcp     open        hosts2-ns
111/tcp    open        sunrpc
443/tcp    open        https
444/tcp    open        snpp
512/tcp    open        exec
513/tcp    open        login
```

```
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
569/tcp    open        ms-rome
1099/tcp   open        unknown
2101/tcp   open        unknown
2161/tcp   open        unknown
4045/tcp   open        lockd
6112/tcp   open        dtspc
6666/tcp   open        irc-serv
7100/tcp   open        font-service
8082/tcp   open        unknown
8083/tcp   open        unknown
32771/tcp  open        sometimes-rpc5
32773/tcp  open        sometimes-rpc9
32777/tcp  open        sometimes-rpc17
32778/tcp  open        sometimes-rpc19
32779/tcp  open        sometimes-rpc21
48265/tcp  open        unknown
55766/tcp  open        unknown
55767/tcp  open        unknown
55768/tcp  open        unknown
55769/tcp  open        unknown
55770/tcp  open        unknown
56492/tcp  open        unknown
56494/tcp  open        unknown
60018/tcp  open        unknown

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=13683 (Worthy challenge)

Sequence numbers: 1A3831F4 1A394172 1A3A473B 1A3B7A53 1A3D0793
Remote operating system guess: Solaris 7
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=3573)
T1(Resp=Y%DF=Y%W=212%ACK=S++%Flags=AS%Ops=NNTME)
T2(Resp=N)
T3(Resp=N)
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=0%IPLEN=70%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)


Nmap run completed -- 1 IP address (1 host up) scanned in 166 seconds
```

Cost Estimates

Product	Function	Available	Approx. Cost
CDRW	Tripwire	www.CDW.com	$200
CheckPoint + Update license	Firewall	Thaumaturgix, NY	$5,500 (need 2)
Ultra 10	Firewall Server	CCNY	$3,500 (need 2)
StoneBeat	Firewall Failover	Thaumaturgix, NY	$12,000
RealSecure + Update License	IDS	www.fishnetsecurity.com	$10,800

References

Mulligan, John (1999). Solaris: Essential Reference
Indiana: New Riders

Pomeranz, Hal; Bishop, Matt, Brotzman, Lee, et al (2000). SANS UNIX Security
Washington DC: SANS Publishing

Northcutt, Steven (1999). Network Intrusion Detection: An Analysts' Handbook
Indiana: New Riders

McClure, Stuart; Scambray, Joel; Kurtz, George (1999). Hacking Exposed: Network Security Secrets and Solutions New York: McGraw-Hill

Anonymous (1998) Maximum Security : A Hacker's Guide to Protecting Your Internet Site and Network New York: Sams

Breton, Chris (1998). Mastering Network Security
California: Sybex