GIAC Enterprises

Unix Security Audit

TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

Abstract

Network security is best accomplished with a methodology that goes step-by-step through a specific system to secure it. This paper provides an example of accomplishing this on a small simple system to learn the basics before moving on to any larger Unix or Linux network implementation in an audit type approach.

This audit was conducted to meet the practical requirement of the Systems Administration, Networking and Security (SANS) Global Incident Analysis Center (GAIC) Unix Security Certification and to evaluate and identify any significant weaknesses in a Unix server for a whimsical ".com" company. As such it involved examination of a Linux file server using the SANS GIAC Step-by-Step approach that performs file server duties for a small internal network. In addition, there are several other systems in this environment including other Linux boxes, some Windows 95/98 workstations and a Windows NT 4.0 server and workstation. All of these computers use a TCP/IP stack.

The major findings of the audit are exposure to at least 5 of the "top 10" security vulnerabilities as published by SANS GIAC (V1.27 8 Sep 2000). They include the use of NFS/RPC services used to launch the famous US Department of Defense Solar Sunrise incident. Two other vulnerabilities are due to the email and/or its underlying protocols like IMAP/POP3 including buffer overflow attacks seemingly so common today.

Finally, there is some exposure due to the use of weak or insecure passwords, especially due to the lack of the use of the `/etc/shadow` file and via the use of Simple Network Management Protocol (SNMP) with "public" or "private" community strings. Another fairly obvious need is to eliminate the infamous 'r' services including rsh, rlogin, telnet, etc. and move to the Secure Shell (SSH) services.

In addition, simple disaster recovery and backup procedures are lacking including console security problems due to the PC hardware CNTL-ALT-DEL reboot sequence, lack of logging in with individual passwords and su'ing in as root, improperly configured PAM module and inadequate system backup and logging methodologies.

Updating the operating system software across the board is probably the single largest security issue on this system. This could be accomplished with the Red Hat `rpm` facilities and the latest distribution of Red Hat, preferable on CD-ROM to insure that no Trojans or other network compromises are induced into the upgrade. **Next, script-based hardening such as Bastille should be accomplished as soon as is possible.**

2.0 Operating System Analysis

2.1 System Configuration

The audited network system is an Intel i586 box with Linux V5.2 that serves as a NFS and SAMBA File Server to other systems located on the internal network (see diagram below). These systems include a small 4-node Linux Cluster for compute bound problems and a variety of other machines including Windows 95/98, Windows NT RRAS/Firewall server and some other older DEC UNIX and VMS systems.

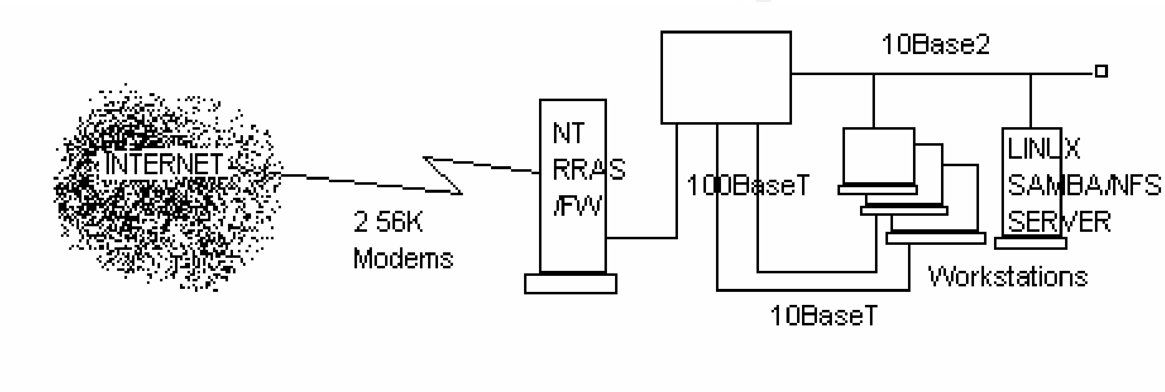

Figure2.1 : Network Map

The external network is currently a Windows RRAS router with limited throughput and dial-on demand connectivity to the Internet with a simple firewall for security. There is a plan to move to new high-speed data connection to the network with a hardware router running at least 128K ISDN or Frame Relay versus the current Microsoft RRAS software solution.

There is limited "defense in depth" in the existing network as only the NT box sits between the Internet and the internal network. Of course, the system is not always connected with this solution. With the strategy to move to an "always on" Internet connection this overall architecture will need to be revisited.

2.2 Files Reviewed

This audit was conducted towards completion of requirements of the SANS Institute's GIAC Level 2 Unix Security Analyst Certification and as such the following configuration files were reviewed.

- /etc/passwd
- /etc/lilo.conf
- /etc/syslog.conf
- /etc/inittab
- /etc/crontab
- /etc/syslog.conf
- /etc/logrotate.conf
- /etc/inetd.conf
- /var/log/secure
- /var/log/messages
- /etc/group
- /etc/shells
- /etc/profile
- /etc/exports
- /etc/services
- /etc/hosts
- /etc/hosts.allow
- /etc/hosts.deny
- /etc/security/limits.conf
- /etc/security/access.conf
- /etc/security/time.conf
- /etc/smbpasswd
- /etc/smb.conf

In addition the following commands were used while investigating the system and the output analyzed.

- ls -lart
- rpm
- ps aux
- ls –la /etc
- ls –la /usr/bin
- ls –la /usr/sbin
- env
- chkconfig --list
- netstat -at
- smbpasswd
- nmapNT

2.3 Vulnerability Testing Tools Used

In addition to the host-based evaluations above, a network-based evaluation was conducted using nmap from the Windows NT platform. It is planned to use crack on the host to verify the strength of the password files as well, but that was not accomplished. The version of nmap used was the Windows NT release from eEye.com. The results are below.

```
Starting nmapNT V. 2.53 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  (
www.insecure.org/nmap/ )

We skillfully deduced that your address is 192.168.42.144
Host samba.42.168.192.in-addr.arpa (192.168.42.143) appears
to be up ... good.
Initiating TCP connect() scan against hack.42.168.192.in-
addr.arpa (192.168.42.143)
Adding TCP port 109 (state open).
Adding TCP port 37 (state open).
Adding TCP port 98 (state open).
Adding TCP port 79 (state open).
Adding TCP port 635 (state open).
Adding TCP port 70 (state open).
Adding TCP port 111 (state open).
Adding TCP port 515 (state open).
Adding TCP port 25 (state open).
Adding TCP port 513 (state open).
Adding TCP port 23 (state open).
Adding TCP port 113 (state open).
Adding TCP port 2049 (state open).
Adding TCP port 110 (state open).
Adding TCP port 21 (state open).
Adding TCP port 139 (state open).
Adding TCP port 143 (state open).
Adding TCP port 514 (state open).
The TCP connect scan took 2261 seconds to scan 1523 ports.

For OSScan assuming that port 21 is open and port 1 is
closed and neither are firewalled.
Interesting ports on hack.42.168.192.in-addr.arpa
(192.168.42.143):
(The 1505 ports scanned but not shown below are in state:
closed)


Port       State       Service
```

```
21/tcp     open        ftp
23/tcp     open        telnet
25/tcp     open        smtp
37/tcp     open        time
70/tcp     open        gopher
79/tcp     open        finger
98/tcp     open        linuxconf
109/tcp    open        pop-2
110/tcp    open        pop-3
111/tcp    open        sunrpc
113/tcp    open        auth
139/tcp    open        unknown
143/tcp    open        unknown
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
635/tcp    open        unknown
2049/tcp   open        nfs

TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)

Sequence numbers: C900216C 5B13227E FE4F54AF ABE98626
34C85F9A AADEC987
Remote OS guesses: Cobalt Linux 4.0 (Fargo) Kernel
2.0.34C52_SK on MIPS or TEAMInternet Series 100 WebSense,
Linux 2.0.35-38

Nmap run completed -- 1 IP address (1 host up) scanned in
2353 seconds
```

Tentative review indicates that for a machine with the NFS/SAMBA file services as its primary function, there are many `inetd` services running that could be eliminated including `ftp`, `telnet`, `gopher`, `finger`, `printer`, etc. The analysis that follows should justify each of these changes, to leave them in, modify the configuration or eliminate them altogether.

2.4 Host Configuration Vulnerabilities Found

This system was loaded from CDROM's included with [PIT98] instead of being downloaded from the Internet. This reduces the chances of getting a Trojan kernel or other operating system components that have been previously compromised.

The first finding that was observed in auditing the host was that the console <CNTL>-<ALT>-<DEL> capability from the default Red Hat installation had not been disabled in /etc/inittab. It was also noted both here and in examining the passwords under Administrative Practices that the sysadmin was logging in using "root" instead of logging in as a user and then su'ing to root [SAN00].

It was recommended that both of these things be changed (see Section 8) and that single user console mode access to the system (Run Level 1) also requires the root password. Physical security to the system was adequate for this Small Office Home Office (SOHO) environment as this file server was kept in a "back office" (actually a separate study off-limits to children and most guests) with limited traffic.

According to the sysadmin the system was configured as "CUSTOM" during the Red Hat installation with only 3 partitions. These where the default /boot (16M), /swap (64M or 2 X physical memory of 32 MB) and the / (root) for the remainder of the disk (approximately 750 MB). All other mandatory filesystems (/var, etc.) are contained along with the SAMBA and NFS mount points in the / partition on this server. One fairly obvious recommendation for continued use of this system as a fileserver is to add a second disk of at least 10GB or more to mount the SAMBA and NFS services on a separate /data or similar type partition.

The next host vulnerability issue was the lack of use of /etc/shadow as the encrypted place to retain the passwords separately from the /etc/passwd file. This capability has additional benefits such as the ability to use other encryption techniques such as MD5 or 3DES for the one-way secure hash for user login and other authentication [SAN00].

A recommendation that was made in this area in consideration of the number of Linux systems in the Linux Cluster on the network and the existence of other UNIX boxes (a Solaris (i386) 8 and a DEC (AXP) UNIX/OSF) is to adopt Caldera Webmin to administer all the systems. This tool runs a 'mini-http" server and can be configured with Secure Shell (SSH) to run a group of cgi scripts that will configure SAMBA, NFS, bind, standard startup scripts, network devices and other things from a web browser like Netscape or Microsoft Internet Explorer.

3.0 Risks From Installed 3rd Party Software

In addition to the standard Red Hat Linux on the server, this company used SAMBA for access between Unix and Windows machines and NFS for Unix-to-Unix file sharing. Both of these tools present a variety of security challenges specific to the business of file sharing. The presence of other significant operating system problems may seem to limit the issues of SAMBA and NFS, but as these are really the whole purpose of this machine. **The NFS and SAMBA components should be viewed clearly as the ongoing risks they are** once the other issues that have been identified in this audit are resolved.

Sun Microsystems Network File System (NFS) has been around for a long time in the UNIX world [NEM89]. As such it has many know vulnerabilities primarily related to its use of Remote Procedure Calls (RPC). RPC authentication uses the IP address and UID/GID of the client, both of which are easily spoofed on a network. File handles used by NFS are `inode` numbers on the server and may also be predicted with some accuracy. Red Hat's pre-V6.0 implementation of NFS (`nfsd`) runs in user space on the server and has several documented exploits. Post-V6.0 Red Hat uses a new kernel mode implementation to remedy some of these exploits, but due to problems in that implementation a kernel version of 2.2.13 or greater is required [SAN00].

There is a secure version of NFS from Sun that relies on Sun's Secure RPC implementation. Even this, however, relies on 56-bit DES encrypted key and as documented in the book "Cracking DES" from Mitch Kapor's (founder of Lotus Development) Electronic Frontier Foundation these one-way keys can be had for $200,000 in hardware and a week of computer time on such a machine [EFF98]. There is also a secure version of NFS client for Linux under development at

http://www.cs.vu.nl/~gerco/SecureRPC/

To access RPC services including NFS, the `portmapper` daemon must be running. This opens another group of exploits based upon the `portmapper` daemon. In should be pointed out that none of the Unix clients that might use the NFS server should also be using the NFS server and should all be running `soft` mounts which eliminates the clients "hanging", unlike the use of `hard` mounts. This, however, presents the problem of data loss if the server goes down and the client times out.

To harden this `UDP` based system the use of `tcp_wrappers` with hosts specified by domain name or IP is recommended. Other important things to check are to insure that no world access is set on the exported filesystems by reviewing the mount points in the `/etc/exports` file and any files in `exports` with world access settings. Use read-only mode unless read-write is required. Finally, the best recommendation is not to use NFS at all if it is not absolutely required. If it is, then to migrate the server to a Sun Solaris platform using Sun's Secure RPC implementation until such a system is available on a Linux box is probably the best bet.

SAMBA is the Unix implementation of Microsoft's Common Internet File System for file and print sharing. Microsoft developed this solution primarily to compete with Sun NFS and Novell file and print servers for that market space. The latest update for SAMBA is V2.0.5a as shipped with Red Hat Linux V6.0. The software has the effect of making a Unix box look like a Windows NT box. An excellent free reference for SAMBA is the O'Reilly book "Using Samba" on the Internet at :

```
http://www.samba.org/samba/oreilly/using_samba/
```

There are four security levels provided by the SAMBA server which is implemented with 2 daemons (`smbd, nmbd`) that provide the file/print and namespace implementations. The security levels are 1) Share – smbd does password authentication at the time of access, 2) User – a look up list-like security check is implemented *by user (preferred)*, 3) Server – a separate SMB password server does the authentication and 4) Domain – a full Microsoft Primary Domain Controller (PDC) does the authentication.

The configuration in this case uses the lowest level of security, the Share level as do most of the other Windows computers in the Microsoft network at this site. It was recommended that 2) above, User level password be implemented. Red Hat should be migrated from V5.2 to V6.2 or later to gain access to the MD5 encryption found in the `/etc/smbpasswd` file or other location set in the `/etc/smb.conf` file as well. Then passwords can be set with the Unix-like command `#smbpasswd` *`<username>`* instead of manipulating the "Windows-like" sections of the `/etc/smb.conf` file with its `hosts allow =` and `hosts deny =` entries. But the sysadmin must be sure to become familiar with the setup in the file

```
/usr/doc/samba-2.0.5a/docs/textdocs/ENCRYPTION.txt
```

before attempting to use this new higher security subsystem.

Another area of investigation related to installed software that was found it this review was the startup scripts for the operating system. This version of Red Hat Linux (V5.2) uses Red Hat's configuration manager Linuxconf V1.12 (Subrev 5) [SIE99]. Some early versions of Linuxconf from Red Hat replace the startup processes with its own version to allow direct manipulation of the scripts in a virus-like behavior[SAN00]. This is another reason to move to a later version of Red Hat (V6.2 or later) as recommended strongly in some other parts of this audit.

4.0 Administrative Practices

The existing guidelines for installing, configuring and maintaining of the Linux systems was said to be directly from [Pitts98]. If so, these procedures did not include the SANS GAIC recommendations found above to disable keyboard boot, use a `/etc/shadow` file or other routine "hardening" techniques. Further findings indicated that use of the Pluggable Authentication Modules (PAM), libraries that control user authentication and allow for simplified upgrade to new encryption algorithms like MD5, 3DES, etc., where minimally configured as well.

Passwords where not written down or very easy to guess by trial and error. Due to limited time, the crack program was not utilized as planned to test for the ease with which the passwords could be guessed. It has been documented that on many systems as many as 50% of passwords can be retrieved using programs like crack within a relatively reasonable amount of time. With other problems like the lack of a `/etc/shadow` file getting the `passwd` file would be much too easy on this particular system. With the limited time available finding these problems was seen as a priority, especially as the sysadmin was aware of many of the common weaknesses in passwords and did seem to be using relatively 'hard to guess' words with mixed alphanumeric and phonetic spellings not found in most dictionaries.

An important use of one of the PAM module configuration files is to modify the `limits.conf` file to limit core dumps to 0 KB in order to limit certain buffer overflow attacks. This configuration file along with the `access.conf` file that controls who is allowed to access things in the operating system and `time.conf` file that control when they have access [SAN00].

System logs where still being kept at the default settings with all log messages in `/var/log/messages` with the exception of the ones generated by the authpriv process that are kept in `/var/log/secure.` Examination of the logs did not indicate any unusual activities (other that the evidence of the nmap port scans used in the audit itself).

Three recommendations from the system logging (`syslogd`) area where made. First, separate the logged messages into 3 files instead of one. The `klogd` should be pointed to a file like `/var/log/kernel` to capture critical messages that might be lost in the bulk of the data in the `/var/log/messages` file. A `/var/log/syslog` file for the `warn` and `err` messages is also a good idea. Finally, modifications to the `/etc/syslog.conf` file to remove entries for the `kern, warn` and `err` levels should be accomplished to prevent duplicate logging of these messages in the `/var/log/messages` file [SAN00].

This has added security benefits, as well as improving the ability of the sysadmin to monitor his system, in that many "Script Kiddies" attack scripts blindly delete the standard log files and might miss the modified system. On a final note, remember that the HUP signal doesn't work well with the klogd. It requires an actual TERM and restart to get going again after modifying its log files in the /etc/syslog.conf file.

Next, it was advised to start the use of log rotation with compression. This is especially critical considering the limited file space on this small server. Some older versions of Linux use shell scripts to run logrotate from cron. This can be very difficult to customize and/or maintain and could produce race conditions that could assist a hacker in compromising the system. While the logs where occasionally reviewed, no consistent pattern for reviewing the logs could be found. To get this site away from this random 'hit or miss' system, one of two scenarios was recommended. The first is below and other is in the subsequent section 5.0.

The first scenario proposed was relatively insecure, but 'doable'. This was to use existing functionality in the /etc/logrotate.conf file to email the compressed logs to the sysadmin. This would get them off the machine and require the sysadmin to at least store them somewhere or use up his mailbox quota. In addition, other backups from the email server and DNS server where already working using this simple technique. The real solution is the syslog server described in the next section, but this was preferable to the existing state of affairs.

Upon examination of the nmap scan in section 2.3, it is apparent that several network components not related to the primary function of this server as a SAMBA and NFS server are available. It is highly recommended that the Internet services daemon (inetd) be disabled for all but the essential needs. As an absolute minimum the "r-services" (rlogin, rsh) should be disabled by entering a "#" to comment them out in /etc/inetd.conf as there are many know exploits that exist for these. Likewise, those services that are critical should be started up in the appropriate init level (/etc/rc.d/init.d) and not dependant on the inetd to start them up when a network connect attempt occurs as is observed with the nmap scan above.

It is possible to audit and correct the inetd services with the netstat command that acts like an "internal nmap". Particularly, on Red Hat Linux the use of netstat –at and chkconfig can accomplish this management task. It is also possible to use another tool, lsof for "List Open Files" to assist in this process. This program was not on the system and would have to be installed from the distribution CDROM.

Ideally, tcp_wrappers should be placed between the hardware level and the inetd. This program is included with most Linux distributions and is found at /usr/sbin/tcpd. Once the tcpd is installed the files /etc/hosts.allow and /etc/hosts.deny need to be properly edited to exclude all but the required access by including the last line ALL : ALL : DENY in the former and only one line in the latter. This is the `ALL: ALL` to deny all access except that explicitly contained in `/etc/hosts.allow`.

5.0 Security Patches and Sensitive Data Storage

Use of Red Hat's update services for both security and functionality upgrades is advisable. This particular system should probably be updated very soon to Red Hat V6.2 or V7.0 (although the sysadmin indicated that he seldom ever used the first release of an operating system, preferring to wait for V7.1 or later). The site `ftp:/updates.redhat.com.6.0/i386` could be used to download `rpm` packages to fix most of the issues know at this point in time. When used with the 'freshen' command `#rpm –F` *`<package-name>`* or Upgrade command `#rpm –Uvh` *`<package-name>`* it will apply the patches or upgrade to the Red Hat system. It should be pointed out that a digital certificate or other means of validating the `rpm` be used before these packages are installed to insure no Trojan is contained within them.

As discussed in Section 4 (above), moving server logs off the file server needs to be accomplished. Ideally***, server logs should be forwarded to a centralized logging server where the logs could be maintained securely.*** The best recommendation, versus just depending on `sendmail`, is clearly to set up a separate `syslog` server. A centralized `syslog` server for all the UNIX machines on the local network could be used to insure that all system logs where saved. If the Red Hat V5.2 on the file server being audited is to log to such a machine, it must be explicitly told to do so by using the '-r' switch in the `/etc/syslog.conf` file [SIE99].

Many attack scripts delete system logs as mentioned above, but if they are sent out to another machine there will be a record of the attack. However, when using this sort of technique it is critical to insure that all the system clocks are synchronized using Network Time Protocol (NTP) or a similar technique.

Perhaps the most straightforward defense for sensitive data is once again the use of good passwords. Passwords should never be written down, should contain both letters and numbers and not easy to guess. As mentioned in the previous section, it seemed that most of this was known to the sysadmin.

6.0 Anti-Virus Software

Lack of any type of anti-virus software to scan the server was observed. However, Windows desktops did use an adequate anti-virus package that was updated fairly regularly. This could be used to scan for know viruses on the shared SAMBA filesystem.

There is still a need, however, for some type of overall anti-virus protection for the UNIX servers and clients using the NFS system. This will serve as a low priority recommendation for that section of the audit.

7.0 Disaster Recovery and Backup Procedures

One of the largest disaster recovery related problem found with the server was the lack of a 'backup' Linux kernel to boot if the original kernel where compromised or merely corrupted from some activity. Further, minimal applications oriented backups where not being adequately accomplished on the server as 'most files resided in multiple places' on the network. This situation should be remedied via the use of some type of tape backup. The only servers in the current network that contain a tape subsystem where the NT RRAS server and some older DEC systems that used all used SCSI tape controllers.

The importance of routine backups on all the servers, including the Linux SAMBA/NFS server was re-emphasized. There where no off-site tapes, however the use of a fire-proof safe was available for on-site tape backups. A strategy to start some type of routine backup of the file server's exported filesystems to back up applications data (as a minimum) was recommended. The availability of both Iomega Zip and Jazz devices in the shop would suggest those removable devices as an alternative to more standard tape backups.

It was recommended highly that a backup kernel be created and `/etc/lilo.conf` modified to insure a bootable Linux was always available on the server. This way if all else failed the sysadmin could enter '`lilo: linux.org`' or whatever the "original" Linux kernel was and recover the system without having to boot from the backup floppy or CD-ROM.

Another important area to backup is the system logs in `/var/log/secure` and `/var/log/messages` and any other system logs on the local or remote machines. Additionally, a recent posting to the SANS Institute Information Security Reading Room (October 2000) suggests an interesting alternate that might be applicable to the centralized `syslog` approach in this particular case. This posting relays the use of an older software package from the now defunct Digital Equipment Corporation called the VAXcluster Console System (VCS). This system runs on a VAX/VMS platform that is in fact already available at this SOHO site and captures and stores serial data from console devices in near real-time into a none-UNIX VAX/VMS Indexed Prolog: 3 file format. This has the effect of making all the UNIX syslogs stored in at least one machine immune to most NT and UNIX known attacks. And the sysadmin at this site has VCS experience, as the author of this posting indicates is the case on his site [JEN00].

No formal Disaster Recovery Plan was evident, although the sysadmin did seem to be familiar with the concepts involved in developing such a plan. It was felt that the current needs of the business did not require more that a minimal network architecture document (similar to the above "generic" Figure 2.1) and list of systems and their application to reconstruct the system.

8.0 Prioritized List of Security Vulnerabilities

8.1 SUBJECT : Non Secure RPC NFS server in use
Risk : High - # 3 on SANS GAIC Top Ten (V1.27 8 Sep 2000)

8.2 SUBJECT : Use of email on the server
Risk : High - #5 and #9 on SANS GAIC Top Ten (V1.27 8 Sep 2000)

8.3 Console Security : <CRTL>+<ALT>+<DEL> on the Red Hat system console
Risk : Medium – high priority due to ease of implementation

8.4 SUBJECT : Insure that an alternate boot kernel is in place for use with LILO :
Risk : Medium – high priority due to ease of implementation

8.5 SUBJECT : Enable /etc/shadow password encryption
Risk : High

8.6 SUBJECT : Change policy to login *<username>* and su to root versus just logging in as root
Risk : Medium (currently only at console or would be High)

8.7 SUBJECT : Disable all inetd services not needed and implement tcp_wrappers (tcpd) in between the hardware and inetd. If possible, completely eliminate inetd.
Risk : High (many vulnerabilities – see nmap scan)

8.8 SUBJECT : Migrate Red Hat to the lastest stable version from the current V5.2 for a variety of reasons including :
Overall Risk : High

8.8.1 Latest security patches
Risk : Very High

8.8.2 SAMBA server version 2.0.5a with MD5 password encryption
Risk : Medium

8.8.3 Elimination of Linuxconf V1.1.2
Risk : Low

9.0 Prioritized List of Recommended Fixes

9.1 Eliminate all of high exposure risks (SANS GIAC "Top Ten") as soon as possible

9.2 Use Red Hat rpm to upgrade the system from V5.2 and get V6.0 and later patches on as soon as possible

9.3 Use Bastille or other hardening scripts on the system as soon as possible

9.4 Implement `tcp_wrappers` and limit `inetd` services offered

9.5 Add compression to `logrotate` procedure and move logging off the file server

9.6 Install `Webmin` to simplify the overall administration of the network

9.7 Separate `syslog` entries into kernel and general log entries

9.8 Install a separate `syslog` server to handle all Unix servers logging needs

9.9 Move to a `Secure RPC` implementation of `NFS`, either Sun Solaris or Linux `Secure RPC` as it becomes available

9.10 Move to a later version of SAMBA with encryption capability and start to use User level SAMBA security versus the Share level currently in use

9.11 Consider the use of the non-UNIX `remotelog` capability in this particular shop due to availability and experience with the VCS software

9.12 Consider some type of UNIX anti-virus for the NFS systems

10.0 References

[SAN00] SANS. Unix Security Analyst Course Materials. GAIC/SANS. 2000.

[JEN00] Jenkinson, John. Using VAX/VMS to Augment Security of a Large UNIX Environment. GAIC/SANS. 2000.

[SIE99] Siever, Ellen. Linux in a Nutshell. O'Reilly. 1999.

[PIT98] Pitts, David and Ball, Bill. Red Hat Linux Unleashed, Third Edition. SAMS. 1998.

[EFF98] Electronic Frontier Foundation. Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design. O'Reilly. 1998.

[NEM89] Nemeth, Evi, Snyder, Garth and Seebass, Scott. Unix System Administration Handbook, First Edition. Prentice Hall, 1989.