Security Analysis Of GIAC Enterprises FTP Gateway

GCUX Practical Assignment
Version 2.0
Option 2 – Consultant's Report
Submitted 2004.01.29

Audit performed by:
Ivar Aarsnes,
Vault Securities
In October 2003

Abstract/Summary

Failing to protect one's data and maintain a high level of availability may often mean make or break in today's highly connected and competitive business market. This becomes increasingly more apparent, as more and more information is stored digitally and made available over different networks. An important realization is that threats also come from inside the company itself, and that a strong first line of defence against the outside is not enough to protect oneself. It is therefore vital to have a steady focus on security in every organization, and on server side security in particular.

In the light of this, GIAC Enterprises hired Ivar Aarsnes of Vault Securities to do a UNIX security audit of the GIAC Enterprises FTP gateway. The server resides on a DMZ between the organizations two main internal networks. It's only objective is to enable users to transfer files between these two.

The objective of this report is to give a detailed technical security analysis of the server in order to identify technical and operational vulnerabilities. This report documents the findings from this audit, and outlines the steps necessary to correct the vulnerabilities discovered.

Table of Contents

Executive Summary

Background
GIAC Enterprises recognizes the importance of computer security, and hired the services of Ivar Aarsnes of Vault Securities in 2003. Being a major player in the fortune cookie sayings market, they realize that protecting their data and keeping it secure from competitors is vital to stay alive in a tough market. In order to find out if their systems are sufficiently secure, they decided to rent the services of an external organization to do a thorough security audit against one of their servers. The customer picked their ftp gateway as the target for this audit. This server is responsible for file transfers between its two main internal networks, and a lot of data passes through this server daily. Some of this data is sensitive, and can lead to great financial losses should they be compromised or stolen by competitors. They scope of this audit was defined to be complete security audit of this server, where potential administrative and configurationally vulnerabilities where uncovered. This audit should also include suggestions on how to correct these vulnerabilities. Network topology and design was not to be included in the audit.

Conclusions
The audit uncovered a number of issues that should be resolved. One of the key findings was the sole use of ftp as file transfer protocol. This is an unencrypted protocol, and an attacker may easily be able to intercept and make use of both data and login/password combos. A switch from the insecure ftp protocol to the secure encrypted SSH protocol is highly recommended.
The audit also uncovered that it is possible to log directly into the server as root should an attacker gain knowledge of the root password. Direct anonymous root access (other then from the local console) should never be allowed. The users should be forced to always identify themselves before becoming root via the su or sudo commands. This both adds an extra level of security and creates an audit trail of root usage.
Another important finding that should be mentioned regards the physical security of the server. The server is placed in a secure facility with strong access control, but it is placed in an unlocked rack. Several people have access to the server room, and everyone with such access will automatically have physical access to the server. Physical access to a server will in most cases enable an attacker to gain full access to the server and its data. The rack containing the server should be physically locked in such a manner that access to it will require a key of its own. This will prevent access to it from unauthorized personnel.

Top priority findings

This is a summary of the top 10 findings from the security audit of the server ftpgw. These are the issues that most urgent need attention. A full listing of the findings can be found in Chapter 3.

1. FTP should be disabled and SCP/SFTP (with scponly as shell) be used instead.
2. Integrity checking of system files should be implemented
3. SSH needs to be properly configured to improve security (no direct root logins, no empty passwords, disable SSH1 support)
4. The server should log events to a central server.
5. Physical security should be improved (lock server rack)
6. The server should be included in normal backup procedures.
7. NTP should be enabled to improve logging and timestamp quality.
8. Disable unneeded services (xinetd, atd and FTP (if #1 is implemented))
9. Create and enforce the use of procedures for account revocation.
10. Remove unnecessary packages

1 Description of System and Audit Methodology

System Description

1.1.1 Hardware

GIAC Enterprises have standardized on Compaq servers for the Intel platform. The model chosen for the server ftpgw is the DL360 G2 model with dual 1.4GHz Intel Xeon CPUs and 2GB RAM. The server has 2x36GB SCSI disks in RAID 1 configuration, and 2x gigabit network interfaces. Only 1 of the network interfaces is connected and in use.

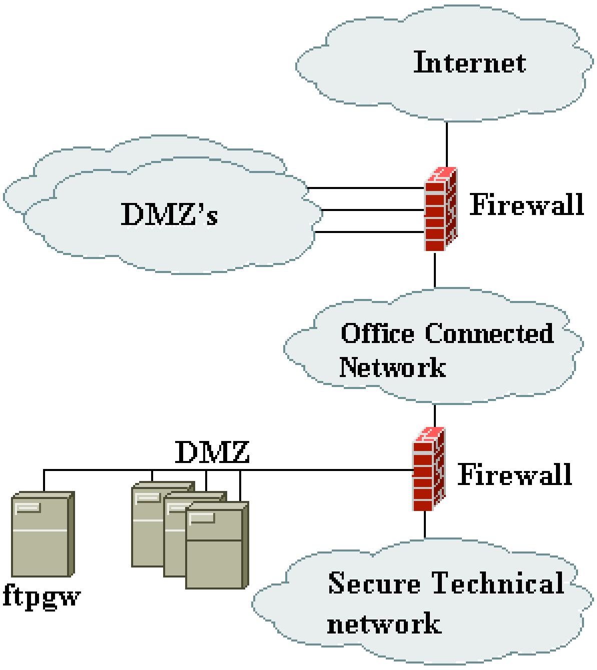
1.1.2 Operating system and version

The operating system and version for the ftpgw server is Red Hat Linux release 8.0 (Psyche) with kernel version 2.4.18-14smp. The system has been installed using the custom install method, with a limited number of packages (a total of 348 see Appendix C for a complete listing). The ftp daemon is vsftpd-1.1.0-1. This daemon is included in the RedHat 8 distribution, and no other software than the RedHat 8 included ones have been installed.

1.1.3 The audited server's role in the organisation

GIAC Enterprises is a large company dealing with online sale of fortune cookie sayings. In such a large company there is a need for different security zones in order to protect company secrets. The ftp gateway in this case is used as a gateway ftp server between the company's office-connected network and the more secure technical network. The office-connected network is where the normal sale, management, HR, and PR departments are located. The technical network is used for research and development of the fortune cookie sayings and the technology used in sale and distribution of these. There are frequent needs for file exchange between these networks (3d party SW, vendor patches, finished products for sale etc), and this is why the server ftpgw has been installed. Its sole purpose is acting as a GW for the transferring of files between these two networks, and it is placed on a DMZ between these two.

1.1.4 Network topology



The server ftpgw is located on a DMZ between the office connected network and the more secure technical network. It is reachable from both networks via ftp (port 20 and 21). SSH (port 22) access is also enabled for administrative purposes, but only from the more secure technical network. No other ports are allowed. There is no direct connection to the Internet.

1.1.5 Risks and concerns

The ftpgw server is used to transfer all sorts of data, including business sensitive and classified data, and a high level of security is therefore vital. Files on this server are automatically deleted after 7 days, and most users delete the files after a successful transfer. This may still be enough time for an attacker to steal or manipulate sensitive data. Since the server is connected to the office-connected network, this means that a wide variety of employees, temps, consultants and other personnel can reach the server. This network also has firewalled Internet access and is reachable for employees working from home or other remote locations. This implies the possibility that an external attacker may exploit these access solutions to gain access the office connected network. The attacker will then be able to reach the ftp gateway.

Although many regard external attackers as the threat, it is important to realize that the real threat may often come from the inside. There have been numerous examples of employees, temps or consultants that for various reasons (financial, espionage, personal vengeance etc.) have caused problems. Relying on protection from external threats may therefore give a false impression that the internal systems are safe. This again stresses the importance that the servers themselves are protected, both from internal and possible external attackers.

1.2 Audit methodology

Various methods have been used in the audit process of the server ftpgw. This chapter details these methods.

1.2.1 Interviews with operation and maintenance personnel

In order to identify company standards and procedures, an interview with the operation and maintenance personnel was conducted. In the interviews the different policies were discussed and the administrative routines and practices uncovered. This gives an understanding and overview of how the server is being operated and maintained from a security point of view, since not everything is documented in writing. Conducting interviews may also discover discrepancies between written documentation and practises.

1.2.2 Physical inspection

The server was physically inspected. This was done to give an impression of the server's physical security. Both the server's physical placement and the server room's security have been taken into consideration (doors, locks, access control, fire alarms etc).

1.2.3 Manual inspection of system files and processes

A manual inspection of several system files and processes was conducted. This was done to detect vulnerabilities and other server information that is not detected by automated tools.

1.2.4 Security checks using automated system tools

The server was checked against known vulnerabilities and misconfigurations using two publicly available tools. These tools are CISscan and Nessus.

1.2.4.1 CISscan – Center For Internet Security's (CIS) Linux benchmark tool

CISscan[1] is a non-intrusive host-based scanner, which scans the system's configuration, and compares it against a database of known vulnerabilities and recommended settings. The result is a score based on the system's overall security on a number based scale. It also provides recommended actions to be taken to improve the system's security. The version used in this audit is a CIS Level-I Benchmark – the prudent level of minimum due cares for operating system security, and is non-intrusive. The scan was run as user root using release 1.1.0 of the Linux version. Results of the scan can be found in Appendix A

[1] CISscan for Linux can be found at: http://www.cisecurity.org/bench_linux.html

1.2.4.2 Nessus

Nessus[2] is a free remote security scanner. The scans were conducted using release 1.4.4 of the Linux version, and run as root using default settings and dangerous plugins disabled. These plugins were disabled in order to minimize the interference with the server's normal operation. The scan was run locally on the machine via an SSH tunnel from a remote console. This was done due to the network topology and configuration. Results of the scan can be found in Appendix B

1.2.4.3 John The Ripper

John The Ripper[3] is an open-source tool for password cracking. This tool was run to detect weak passwords that were easily guessable. Abusing weak passwords are a frequently used way of gaining unauthorized access to servers. The tool was run on a merged password/shadow database. This merging was done using the utility unshadow, which is provided with the distribution of John The Ripper. The program was run using the "-single" option in order to use information obtained from the password database when guessing passwords.

[2] Nessus can be found at: http://www.nessus.org/intro.html
[3] John The Ripper can be found at http://www.openwall.com/john/

2 Detailed Analysis

2.1 Operating system vulnerabilities

In order to detect Operating system vulnerabilities a range of tools and manual inspection of system files where used. This chapter details the use and results of these.

2.1.1 Results from CISscan

The CISscan program was run as root using the command "/usr/local/CIS/cis-scan". To increase readability in the report, only negative findings have been included and commented here. The full output from the scan is included in Appendix A

Negative: 1.2 sshd_config parameter Protocol is not set
This option should be set to "Protocol 2" in /etc/ssh/sshd_config. The SSH version 1 protocol is known to have security vulnerabilities[4], and use of version 2 should be enforced bye the SSH daemon. This may require clients to be updated (if clients only supporting SSH1 are in use), but this should not be a problem in this scenario.

Negative: 1.2 sshd_config parameter PermitRootLogin is not set.
This option should be set to "PermitRootLogin no" in /etc/ssh/sshd_config. Root should never be allowed to log directly in to a system. Users should be forced to use sudo or su to root in order to keep an audit trail on who has issued commands as root. This also requires that the users wishing to execute commands on the server as root already has an account with a valid shell. This reduces the risk of the server being compromised, should someone illegally get the root password.

Negative: 1.2 sshd_config parameter PermitEmptyPasswords is not set.
This option should be set to "PermitEmptyPasswords no" in /etc/ssh/sshd_config. Requiring the user to provide a secret password in order to authenticate is one of the most basic level of authentication. No users should be allowed to log in without supplying a valid password.

*Negative: 1.2 ssh_config must have 'Protocol 2' underneath Host **
The same as for sshd_config applies here. It is good practise to set this option to "Protocol 2" in /etc/ssh/ssh_config. This is however not critical in this particular case, since the firewall blocks ssh traffic originating from the server[5].

Negative: 2.3 ftp not deactivated.
Since this is an ftp server, this is a false error and can be ignored. This check is for servers not used as ftp-servers, since good practise is to disable all unnecessary services. See Chapter 3.1 for recommendation regarding switch to scp/sftp.

[4] As quoted from http://www.openssh.org/security.html *"OpenSSH has the SSH 1 protocol deficiency that might make an insertion attack difficult but possible. The CORE-SDI deattack mechanism is used to eliminate the common case. Ways of solving this problem are being investigated, since the SSH 1 protocol is not dead yet."*

[5] Only ssh connections to the server from the secure technical network is allowed. All ssh client traffic originating from inside the DMZ and out towards any network is blocked.

Negative: 3.2 xinetd is still active.
vsftpd is run from xinetd, and thus xinetd is active. Since no other services are run from xinetd, it is recommended that xinetd be disabled, and vsftpd run as a standalone daemon.

Negative: 3.9 NFS script autofs not deactivated.
This server does not access NFS shares, and this should therefore be disabled. Disable this using the command "chkconfig –level 345 autofs off"

Negative: 3.21 Kudzu hardware detection program has not been deactivated.
Kudzu has no mechanisms for authentication, and will automatically add, remove and/or reconfigure devices based on hardware changes. This may enable personnel with physical access to the system to configure and add new devices to the system, which is a security risk. It is recommended to disable this daemon to prevent this. Disable this using the command "chkconfig –level 345 kudzu off"

Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/secure_redirects should be set to 0[6].
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/secure_redirects should be set to 0.
The secure_redirects parameter controls if the server should honour ICMP redirects coming only from routers defined as default gateway. Since the server only has one defined default gateway, this should be disabled. Se also general comments regarding redirects below. This should be disabled by setting "net.ipv4.conf.all.secure_redirects = 0" in /etc/sysctl.conf and rebooting.

Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_redirects should be set to 0.
The accept_redirects parameter controls if the server should honour ICMP redirects coming from any hosts. This should normally always be disabled, as it leaves the server vulnerable for man-in-the-middle attacks. ICMP redirects enables a hostile server to tells the victim server to route all its packets through it by posing as a better route. This enables the attacker to monitor, sniff and potentially manipulate traffic from the victim server. This should be disabled by setting "net.ipv4.conf.all.accept_redirects = 0" in /etc/sysctl.conf and rebooting.

Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_source_route should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_source_route should be set to 0.
The accept_source_route parameter controls if the server should accept source routed packets. Such packets specify explicitly how packets should be routed, and is normally only used with malicious attempt. Attackers can use this to spoof source addresses in order to impersonate other hosts (i.e. in order to divert traffic or abuse trust relationships) or to get around TCP wrappers or other packet filters.
This should be disabled by setting "net.ipv4.conf.all.accept_source_route = 0" in /etc/sysctl.conf and rebooting.

Negative: 4.1 /proc/sys/net/ipv4/tcp_max_syn_backlog should be at least 4096 to handle SYN floods[7].

[6] All settings from negative 4.1 is explained at: http://www.securityfocus.com/infocus/1711
[7] This setting is further explained at http://www.die.net/doc/linux/man/man7/tcp.7.html

The tcp_max_syn_backlog parameter sets the maximum number of queued connection requests, which have still not received an acknowledgement from the connecting client. The kernel will begin dropping requests when this number is exceeded, thus blocking traffic. This parameter should be increased to 4096 in order to better handle SYN floods[8]. Set the parameter "net.ipv4.tcp_max_syn_backlog = 4096" in /etc/sysctl.conf and reboot in order to increase this to 4096.

Negative: 4.2 /proc/sys/net/ipv4/conf/eth0/send_redirects should be set to 0.
Negative: 4.2 /proc/sys/net/ipv4/conf/lo/send_redirects should be set to 0.
This machine is not a router, so there should never be any need to send any ICMP redirects. Disable this by setting the parameter "net.ipv4.conf.all.send_redirects = 0" in /etc/sysctl.conf and rebooting.

Negative: 5.2 /etc/vsftpd.conf should have log_ftp_protocol set to yes.
Commands sent to the ftp server should be logged. This may uncover attempts to exploit the server (i.e. by sending bogus commands), and also gives an audit trail of use of the ftp service.Enable this setting by setting "log_ftp_protocol=YES" in /etc/vsftp.conf.

Negative: 5.2 /etc/vsftpd.conf should not have xferlog_std_format set to yes.
If this setting is set to yes, the ftp server will log transfers in a wu-ftpd compatible log format. This only logs file transfers, and not connections. This parameter should be set to no in order to get the best logging. Disable this setting by setting "xferlog_std_format=NO" in /etc/vsftpd.conf.

Negative: 6.1 /var is not mounted nodev.
Negative: 6.1 /ftproot is not mounted nodev.
Negative: 6.1 /boot is not mounted nodev.
Filesystems that doesn't normally have devices in them should be mounted with the nodev option[9]. This prevents users from mounting unauthorized devices in filesystems that should not contain devices. Chrooted environments may be required to have devices in those filesystems. In this case no chroot environments are in use, and thus the above filesystems should be mounted nodev. This is achieved by having the parameter "nodev" in the fourth column in /etc/fstab for the entries for /var, /ftproot and /boot.

Negative: 6.2 Removable filesystem /mnt/cdrom is not mounted nosuid.
Negative: 6.2 Removable filesystem /mnt/cdrom is not mounted nodev.
Negative: 6.2 Removable filesystem /mnt/floppy is not mounted nosuid.
Negative: 6.2 Removable filesystem /mnt/floppy is not mounted nodev.
The above applies to removable filesystems as well. None of these should contain devices. We should not honour the SUID bit on files from removable media either, as an attacker could easily start a rootshell if he is allowed to mount media with such files on it. To prevent such abuse, put the parameters "nodev,nosuid" to the fourth column in /etc/fstab for the entries for /mnt/cdrom and /mnt/floppy

[8] SYN attacks is a DOS attack, where an attacker floods the server with SYN packets in order to fill the servers back log queue and make the kernel drop packets. This in turn prevents normal traffic.
[9] Further information can be found in the document LinuxBenchmark.pdf from http://www.cisecurity.org/bench_linux.html

```
Negative: 6.3 PAM allows users to mount removable media: <floppy>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <cdrom>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <pilot>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <jaz>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <zip>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <ls120>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <camera>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <memstick>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <flash>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <diskonkey>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rem_ide>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rio500>.
(/etc/security/console.perms)
```

There is no need for users on the console to be able to mount removable media on this server. Such access should therefore explicitly be denied. Although this server is located in a secure server room away from normal users, it is still good practice to disable this. This is done by adding a “#” in front of the entries for permission definitions for the above media in /etc/security/console.perms. (Thus making the lines look similar to this: “#<console> 0600 <zip>   0660 root.disk”)

```
Negative: 7.3 User mailnull is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User nscd is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User apache is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User rpcuser is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User gopher is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User rpc is not present in /etc/vsftpd.ftpusers
```

These system-users should have locked accounts and there should never be any need to log in via ftp to the server as these. Ftp access to these accounts should therefore be denied. Denying ftp login to users is done by adding the username (on a separate line) to the file /etc/vsftpd.ftpusers.

```
Negative: 7.5 Couldn't open cron.allow
Negative: 7.5 Couldn't open at.allow
```

The files cron.allow and at.allow specifies which users are allowed administrative access to the crontab and at commands to schedule jobs to be run at intervals. These files should be created and include only the users that need to do this. In this case only root should be allowed such access. Enable this access control by doing the following commands:
“rm –f /etc/at.deny /etc/cron.deny”

“echo root > /etc/cron.allow”
“echo root > /etc/at.allow”
“chown root:root /etc/cron.allow /etc/at.allow”
“chmod 400 /etc/cron.allow /etc/at.allow”
This ensures that only root be allowed administrative access to at and crontab.

Negative: 7.6 The permissions on /etc/crontab are not sufficiently restrictive.
/etc/crontab should only be accessed by the cron daemon (running as root) and the crontab command (SUID root). There is therefore no need for users to access this file, and access to it should be restricted. Restrict access to it by setting it to only be readable by root. This is done with the command “chmod 400 /etc/crontab”.

Negative: 7.7 No Authorized Only banner for vsftpd in file /etc/vsftpd.conf.
A warning banner should be added to the ftp server. This should state that the server is for authorized access only and activity may be monitored and recorded. The organizations legal counsel should review the contents of this banner before implementing it. Such banners should ideally be standardized and used across all servers in the organization. The banner is enabled by using the “ftpd_banner=” parameter in /etc/vsftpd.conf.

Negative: 7.8 xinetd either requires global 'only-from' statement or one for each service.
Xinetd is only used for serving vsftpd, and should as stated above be disabled. If xinetd is disabled and vsftpd run standalone, no further changes are needed to xinetd’s config.
If xinetd is not disabled and vsftpd is left running from it, changes restricting access should be implemented. Use the “only_from” parameter in the file /etc/xinetd.conf file, and include the address ranges that ftp is allowed and expected from. (i.e. setting “ only_from = 192.168.0.0/24” restricts access to addresses from the192.168.0.xx IP-address segment)

Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/11.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty11.
Root should never be allowed to log in anonymously from any other terminal than the local console. User accessing the root command should be forced to use sudo or to su to root from a normal user account. This preserves an audit trail for root access, and requires that users have a normal user account in order to become root, and not just knowledge of the root password. In case of emergency root should be allowed to access the physical console anonymously, but this console should be well protected. Remove the lines above from /etc/securetty to improve security.

Negative: 7.11 /etc/inittab needs a /sbin/sulogin line for single user mode.

In order to prevent users with physical server access (not to big an issue in this case, but it should be implemented anyway) from booting into a single user root shell, an entry for /sbin/sulogin in run level S should be added to /etc/inittab.
Adding the following line after the "initdefault" entry does this:

"~~:S:wait:/sbin/sulogin"

This setting presents the user with the following prompt when booting into single user mode (and thus requiring knowledge of the root password before being handed a shell":

"Give root password for system maintenance
(or type Control-D for normal startup):"

Negative: 8.1 bin has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 daemon has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 adm has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 lp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 mail has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 news has a valid shell of /bin/sh. Remember, an empty shell field in /etc/passwd signifies /bin/sh.
Negative: 8.1 uucp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 operator has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 games has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 gopher has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 ftp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nobody has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpc has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 vcsa has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nscd has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 sshd has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpm has a valid shell of /bin/bash.

Negative: 8.1 mailnull has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 smmsp has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpcuser has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 pcap has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 apache has a valid shell of /sbin/nologin. Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
These accounts are all non-human accounts that should both be locked and have an shell not included in the list of valid shells (/etc/shells). It is advisable to set the shell to /dev/null instead of /sbin/nologin. This makes it more difficult for an attacker to replace the shell, because if he replaces /dev/null with a shell, the systems will stop working[10]. /sbin/nologin should also be removed from /etc/shells.

Negative: 8.3 User a34432 should have a minimum password life of at least 7 days.
Negative: 8.3 User a34432 should have a maximum password life of between 1 and 90 days.
Negative: 8.3 User a12566 should have a minimum password life of at least 7 days.
Negative: 8.3 User a12566 should have a maximum password life of between 1 and 90 days.
(…)List has been truncated; all normal users are listed here (…)
Strong passwords, which cannot be altered, are issued bye the system administrators. The passwords have therefore been set to never expire. If a user needs a new password, a new strong one is reissued. This has both an upside and a downside to it. The good news is that the users always have strong passwords that are difficult to guess and/or crack. The bad news is that since ftp is a clear-text protocol. This means that passwords are susceptible to sniffing. An attacker who can intercept/sniff traffic to the server can read login/password combos in clear-text, no matter how difficult they are to crack or guess, An attacker who does this, or otherwise gains knowledge of login/password combos, will then have access to a user account with a password that doesn't expire. Since users doesn't have a valid shell, it is also impossible for them to change their password. See Chapter 3.1 for further discussions regarding FTP and SCP/SFTP.

Negative: 8.3 /etc/login.defs value PASS_MAX_DAYS = 99999, but should not exceed 90.
Negative: 8.3 /etc/login.defs value PASS_MIN_DAYS = 0, but should not be less than 7.
These parameters control the number of days allowed between password changes: Since the users are issued passwords by the administrators, these values have been set this way. As long as users are unable to change their passwords, these settings are fine. If changes are made later, that enables the users to change passwords, these parameters should be changed to ~45 and ~7 respectively.

Negative: 8.3 /etc/login.defs value PASS_MIN_LEN = 5, but should be at least 6.

[10] As suggested in the document LinuxBenchmark.pdf from http://www.cisecurity.org/bench_linux.html

This parameter controls the minimum allowed password length. In order to make brute-forcing passwords more difficult, this parameter should be set to 7. Change this by setting “PASS_MIN_LEN=7” in “/etc/login.defs”

Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be stronger to block group-read/write/execute.

These setting are global, and sets the default umask value for all users. Since all user-directory have 700 permissions on them (restricting access to only the user himself/herself), this isn’t a big deal in this case. Good practice is to set these parameters correct anyway. Set the umask parameter to 077 (in the files above) in order to block group and world read/write/execute access.

Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger to block group-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be stronger to block world-read/write/execute.

Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be stronger to block group-read/write/execute.

These are the same settings as above, but for roots account only. Set the umask parameter to 077 in the respective files.

Negative: 8.11 Coredumps aren't deactivated.

Since this isn’t a development server where core dumps are actively being produced and used, these should be disabled. Coredumps may contain pieces of sensitive information (i.e. entries from the password and shadow file), which can be exploited by potential attackers. These files may also be used in DOS (DenialOfService) attacks against this server. Coredump-files may be very large, and may therefore fill the filesystems. This may cause the server to crash or otherwise impact it's services. Set the parameter "ulimit -c 0" in /etc/profile to disable coredumps for all users (users may override this parameter in their local files (.bashrc, .cshrc, tcshrc etc.) but disabling it by default may at least give some protection)

Negative: 6.6 Non-standard world-writable file: /var/www/html/signatur.log
Negative: 6.6 Non-standard world-writable file: /var/www/html/guestbook.html
World writeable directories can be abused in several ways (i.e. to bypass disk-quotas on user directories, by putting files elsewhere), and should not be used at all if avoidable. In this case the entire /var/www directory is not needed, and can safely be deleted, since there is no web-server running on this server Remove the directory /var/www with the command "rm –rf /var/www".

Ending run at time: Fri Oct 17 14:13:16 2003
Final rating = 6.31 / 10.00
This is the end result of CISscan. This score will be considerably improved if the changes above are implemented.

2.1.2 Results from Nessus scan

The scans were conducted using release 1.4.4 of the Linux version, and run as root using default settings and dangerous plugins disabled. These plugins were disabled in order to minimize the interference with the server's normal operation.
The firewall protecting the server has been confirmed to block all other ports than ftp from the office connected network, and all other then ftp and ssh from the technical network. The scan was run locally on the machine via an SSH tunnel from a remote console in order to detect all anomalies on the server. By doing this we are able to get results for services that may be exposed if the FW becomes compromised or misconfigured, or in the event that other servers on the same DMZ should be compromised. This method of scanning also prevents the FW and potential IDS probes between the scanning host and the scanned host from generating alerts and extensive logging, which in turn may impact normal operations. The scan was done as none-intrusive as possible.
Below are the results from the scan. The identification that the SSH and ftp service is running, has not been included and commented, since these are meant to be running. The full scan report is included in Appendix B

The scan found a total of 11 security holes, classified in this manner:
high severity : 1
low severity : 8
informational : 2

Only 2 open ports where found, which confirms that no other services are running then the required ones. These are the same services that are opened in the FW. The ports that Nessus found open was:

ssh (22/tcp)
ftp (21/tcp)

The high severity security hole that was found was this:

Service: ssh (22/tcp)
Severity: High

You are running a version of OpenSSH which is older than 3.7.1
Versions older than 3.7.1 are vulnerable to a flaw in the buffer management functions which might allow an attacker to execute arbitrary commands on this host.

An exploit for this issue is rumored to exist.
It appears that this server runs an OpenSSH version with a critical security hole. (Addressed in CERT® Advisory CA-2003-24 Buffer Management Vulnerability in OpenSSH[11]) Further investigation showed that this was not so, and that Nessus gives a false positive. Nessus only checks the version number and alert if the version is below 3.7.1. RedHat have backported the security patch to version 3.4, and released version 3.4.p1-7, which is unaffected by the bug.

Nessus confirms that this may be a false positive by stating in the scan that:
"Note that several distribution patched this hole without changing the version number of OpenSSH. Since Nessus solely relied on the banner of the remote SSH server to perform this check, this might be a false positive."

Further information can be found in advisory RHSA-2003:279-17[12] from RedHat. This finding was confirmed to be a false positive by running the command "rpm –qa | grep ssh". The result from this command was:

```
openssh-3.4p1-7
openssh-server-3.4p1-7
openssh-clients-3.4p1-7
```

These are the patched and secure OpenSSH versions, which mean that the server isn't affected by this bug in OpenSSH.

Service: ftp (21/tcp)
Severity: Low

Remote FTP server banner :
220 ready, dude (vsFTPd 1.1.0: beat me, break me)
This indicates that there is a banner reporting the software version of the vsftp daemon. This information can be helpful for an attacker. If there should be discovered a bug that can be exploited in vsftp, the attacker will be able to see if this version can be exploited. The banner can be changed by using the "ftpd_banner=" parameter in /etc/vsftpd.conf. This banner should be a short authorized usage only banner. See discussion concerning vsftp banner in chapter 2.1.1

[11] This advisory can be found at http://www.cert.org/advisories/CA-2003-24.html
[12] This advisory can be found at: http://rhn.redhat.com/errata/RHSA-2003-279.html

```
Service: ssh (22/tcp)
Severity: Low

You are running OpenSSH-portable 3.6.1p1 or older.

If PAM support is enabled, an attacker may use a flaw in this version
to determine the existence or a given login name by comparing the times
the remote sshd daemon takes to refuse a bad password for a non-existant
login compared to the time it takes to refuse a bad password for a
valid login. An attacker may use this flaw to set up a brute force attack against
the remote host.
```

PAM support is not enabled on the server according to /etc/ssh/sshd_config. This vulnerability does therefore not apply here.

```
Service: ssh (22/tcp)
Severity: Low

The remote SSH daemon supports the following versions of the
SSH protocol :

  . 1.33
  . 1.5
  . 1.99
  . 2.0

These protocols are not completely cryptographically safe so they should not be
used.
```

As mentioned earlier in chapter 2.1.1, the SSH daemon should be configured to only support the more secure SSH 2 protocol. This is accomplished by setting the parameter “Protocol 2” in /etc/ssh/sshd_config.

```
Service: ssh (22/tcp)
Severity: Low

You are running OpenSSH-portable 3.6.1 or older.
There is a flaw in this version, which may allow an attacker to bypass the access
controls set by the administrator of this server.
```

This is not an issue in this case since access control based on addresses isn’t used.

2.1.3 Results from John The Ripper

John The Ripper was run against a merged password/shadow database from the server ftpgw. It was run with the command line “./john -single pw.txt”.
This gave the following output (login/password has been altered):

```
Loaded 169 passwords with169 different salts (FreeBSD MD5 [32/32])
a3648         (a3648)
guesses: 1  time: 0:00:00:50 100%  c/s: 3432  trying: amike75
```

The system administrators is responsible for issuing strong non user-changeable passwords, and it was therefore not expected find any passwords in this way. It turned out however that a member of the administrative staff had himself a guessable password. This emphasizes the need for all to adhere to policies. No other passwords were discovered using this tool. More passwords may have been cracked using brute force and letting it run for extended amounts of time. This was not done since the reason for the scan only was to detect weak passwords, and not brute force them.

2.1.4 Manual inspection of files and processes

A listing of running processes was made to see if any unnecessary or suspicious processes were running. The command “ps –ef” gave this listing:

```
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0  2003 ?        00:01:06 init
root         2     1  0  2003 ?        00:00:00 [migration_CPU0]
root         3     1  0  2003 ?        00:00:00 [migration_CPU1]
root         4     1  0  2003 ?        00:00:00 [keventd]
root         5     1  0  2003 ?        00:00:00 [ksoftirqd_CPU0]
root         6     1  0  2003 ?        00:00:00 [ksoftirqd_CPU1]
root         7     1  0  2003 ?        00:02:39 [kswapd]
root         8     1  0  2003 ?        00:00:00 [bdflush]
root         9     1  0  2003 ?        00:00:14 [kupdated]
root        10     1  0  2003 ?        00:00:00 [mdrecoveryd]
root        18     1  0  2003 ?        00:00:07 [kjournald]
root        95     1  0  2003 ?        00:00:00 [khubd]
root       167     1  0  2003 ?        00:00:00 [kjournald]
root       168     1  0  2003 ?        00:00:12 [kjournald]
root       169     1  0  2003 ?        00:00:06 [kjournald]
root       458     1  0  2003 ?        00:00:10 syslogd -m 0
root       462     1  0  2003 ?        00:00:00 klogd -x
root       541     1  0  2003 ?        00:00:03 xinetd -stayalive -reuse -pidfile
/var/run/xinetd.pid
root       551     1  0  2003 ?        00:00:09 crond
daemon     569     1  0  2003 ?        00:00:04 /usr/sbin/atd
root       577     1  0  2003 tty2     00:00:00 /sbin/mingetty tty2
root       578     1  0  2003 tty3     00:00:00 /sbin/mingetty tty3
root       579     1  0  2003 tty4     00:00:00 /sbin/mingetty tty4
root       580     1  0  2003 tty5     00:00:00 /sbin/mingetty tty5
root       581     1  0  2003 tty6     00:00:00 /sbin/mingetty tty6
root      1716     1  0  2003 ?        00:00:00 login -- root
root      1730  1716  0  2003 tty1     00:00:00 -bash
root     17008     1  0  2003 ?        00:00:05 /usr/sbin/sshd
root     12261 17008  0 Jan15 ?        00:00:00 /usr/sbin/sshd
root     12263 12261  0 Jan15 pts/1    00:00:00 -bash
root     18780 17008  0 14:42 ?        00:00:00 /usr/sbin/sshd
a3414    18782 18780  0 14:42 ?        00:00:00 /usr/sbin/sshd
```

```
a3414   18783 18782  0 14:42 pts/0    00:00:00 -bash
root     18817 18783  0 14:42 pts/0    00:00:00 su -
root     18818 18817  0 14:42 pts/0    00:00:00 -bash
root     18858 18818  0 14:42 pts/0    00:00:00 ps -ef
```

There is only a small number of processes running as shown above. xinetd (PID 541) can be stopped by running vsftpd as a standalone process (See comments for Negative 3.2 in chapter 2.1.1).
The atd daemon can also be disabled. At commands are not being used, and if scheduled jobs are necessary in the future, cron can be used for this. Disable this service with the command "chkconfig –level 345 atd off".
No other suspicious or unnecessary processes seem to be running.

2.2 Security patch installation/management

There is no central patch repository, and no central patch management and installation tool is being used. No written policies or procedures regarding this have been discovered. The administrators regularly download and install security patches as they are released from RedHat. Based on the scans of the server and interviews with the system administrators, this seems to work fairly well, as the server was patched and up to date.
Given that the organization maintains several servers, a centralized patch repository on a central secured server would be advisable. A defined group of persons should be responsible for monitoring security bulletins and vendor errata. These should ideally also test new patches before they are implemented in the production environment.
There should also be defined procedures to inform all system administrators when new patches become available, and when relevant security holes are found. This would ensure that all system administrators becomes aware of new patches, and are quickly able to install them as they already can be found inside the organization. It also exists tools for patch installation and management that enables servers to be patched directly from a centralized server. A detailed overview of such a set up is beyond the scope of this document, but further investigation regarding this is advised.

2.3 Configuration vulnerabilities

Results detected from automated scans are not included here, since they are detailed in chapter 2.1.1.

2.3.1 Installation

The server has been installed with the custom install option, and a limited number of packages have been installed. There are still a number of unused packages that are installed using this option, and these should be removed. It is recommended that the included list of packages be examined for unnecessary packages, and that these be removed. One should be careful to examine dependencies, so that packages necessary for other needed packages aren't removed. The rpm command normally keeps track of this itself via the rpm database, but it should be taken into consideration nonetheless. A full listing of currently installed packages is included in Appendix C

2.3.2 Logging

The server is not configured to log to a central log host. If an attacker is able to compromise the server, he will be able to modify the system logs. This can hide and destroy evidence of attacks and break-ins. If the server logs to a central server in addition to the local server, a much better audit trail is kept. It is then possible to discover the attack, since the attacker is unable to modify the logs on the log host. Logging to a central server also has the benefit that logs can be monitored and analysed centrally. This can give a faster response time in the event of attacks, and makes correlating events among multiple hosts easier. It is advised that such a solution is put into place, but the details concerning the configuration of such a server is beyond the scope of this report.
The server ftpgw should be configured to log event to central server by adding these line to the file "/etc/syslog.conf":

```
*.crit;*.err;*.emerg;auth.warning                              @central.log.host.com
```

This will enable logging of all critical, emergency and error messages to another server. It will also log authentication warnings in addition to this.

2.3.3 Time synchronization

The server is not configured to use NTP. Time synchronisation should be used on all servers. By ensuring that all clocks are in sync, correlating and reconstructing events based on system logs would be much easier. Being able to pinpoint the exact time of attacks on servers can make a world of difference sometimes. NTP for RedHat 8 is included in the RedHat 8 distribution, and can be downloaded from this link:

ftp://sunsite.uio.no/pub/unix/Linux/RedHat/redhat/8.0/en/os/i386/RedHat/RPMS/ntp-4.1.1a-9.i386.rpm

Updates may be available after this report is written. It is therefore advisable to check the update directories for updated versions first. Updates can be found here:

ftp://sunsite.uio.no/pub/unix/Linux/RedHat/updates/8.0/en/os/

This package should be installed, and the server configured to update from 2 or 3 ntp servers.

2.4 Risks from installed third-party software

No third-party software has been installed on the server, and whence no risk introduced from such.

2.5 Administrative practices

2.5.1 User creation and removal

There exist a formal procedure for creating users. This include all actions from when a users needs access to how the user should be created. This procedure is being actively used, and works well. There is however no formal procedures describing action when a user no longer requires access. It should be established a better link between HR, head of departments and the systems administrators. Today there is no automatic removal of users. Accounts can be inactive for quite sometime before they are removed, if they are removed at all. This is because no one ever informs the system administrators that the user no longer requires access. Today the head of the different apartments are responsible for ordering access for his or her employees. They should also be responsible for ordering the removal of users in the event that they no longer require access. HR should also be responsible for sending lists of people that has left, in order to make sure that users are removed from all servers. This process should be formalized in written procedures and policies.

2.6 Identification and protection of sensitive data on the host

The server is sometimes used for transferring sensitive data between the networks. There are no mechanisms or policies for protecting and/or identifying sensitive data on the host. Data stored here are only stored temporary, and all data are treated equally. Standard UNIX rights are used to protect data, and all user directories have permissions set to 700 (rwx------).

There are no policies for how sensitive data should be handled on the host. Written policies should be in place, which ensures that sensitive data is deleted immediately after a successful transfer. This will reduce the potential exposure time on the host if it is compromised (except in the case where and attacker automatically copies data entering the server). The policies should also include a definition of sensitive data.

2.7 Protection of sensitive data in transit over the network or Internet

There is no protection of sensitive data in transit over the network. The protocol for file transfers is ftp, and whence all communication is done in clear-text. The lack of strong authentication and encryption of traffic, makes file transfers susceptible for possible data-manipulation and –theft. It also makes it possible to sniff login/password combos that enable an attacker full access to accounts used for transferral of such data. Traffic originating from Internet is encrypted using a protected VPN from the users PC to the inside of the office-connected network. Traffic from this point to the ftpgw server is unencrypted as the rest of the traffic.

Traffic should be secured and encrypted using SSH instead of FTP for all file transfer. All traffic between the client and server should be encrypted. This will have a number of benefits.

- man-in-the-middle attacks become difficult/impossible (unless he has access to the servers and clients private keys) since strong authentication is used.

- Data theft and manipulation becomes difficult/impossible since all data is encrypted in a strong and secure manner
- Sniffing of login/password combos becomes difficult/impossible since this information no longer is sent in clear-text, but is encrypted in a strong and secure manner.
- By encrypting all traffic, it will be difficult for an attacker to know which traffic is sensitive. He/she will need to record all traffic and try to decrypt everything in order to discover sensitive data. Since everything is securely encrypted, this will make the job close to impossible. This is at best extremely time consuming, especially since there is no way of knowing which data is relevant for him/her.

2.8 Physical Security

2.8.1 Physical Access control and security

The server is located in a dedicated specialized server room on the 5th floor (in a 7-story building) together with several other systems. Power to the server is protected using UPS (using both batteries and diesel powered generators.), which is tested regularly to insure proper operation. The room also has both raised floor and an extra steel roof to provide protection from water, which may enter the room from leaks elsewhere in the building or from fire extinguishing. Raised floor also hides the cabling to the different racks. The cabling is not placed directly on the floor beneath the raised tiles, but on special elevated gateways that runs directly beneath the tiles. The fire detection system is a laser based particle detector system, with probes over every rack and several other places in the room. There is no automatic fire extinguishing system in the room. It was once protected using a Halon gas fire suppression system, but due to recent laws prohibiting the use of Halon gas, this has been removed. It has not been replaced by another system. Entrance to the building requires the use of a personalized magnetic key card and an access code. Three of these doors must be passed in order to reach the server room. This room has two steel doors (on opposite sides of the room), of which only one can be opened from the outside. The entrance door to the server room has the same lock system as the other doors. The use of personalized magnetic key cards and access code provides both excellent access control and an audit trail of people entering the room. The servers that control the electronic locks are located in another secure building, and cannot be manipulated from inside the room. There are several windows in the room, but these are all bullet proof and cannot be opened. The brick walls are thick and solid. No cameras, microphones or other surveillance equipment are used in the room to record activity besides the access control to the room. The overall security and protection of the server room is very good.

The server itself is placed in a lockable rack, but this lock is not in use. This is a potential threat since all personnel with access to the room will have physical access to the server. Access to the room is granted to several people besides the systems administrators responsible for the server, making this a real threat. An attacker may either boot into the system from alternate media or boot into single user as root without password, thus gaining root access to the server.

2.9 Access controls

2.9.1 Physical Access control

Physical access control is by the use of personalized magnetic key cards and access codes. This control only applies to gaining access to the server room. Once inside the server room, there is no physical access control. The server rack should be locked, and thus enforcing a final level of physical access control. Se further comments in chapter 2.8.1

2.9.2 Privileged accounts access

There are no other privileged accounts besides the root account. The root account can be reached directly via SSH and from the system console. Enable direct root access over the network is strongly discouraged. This severely limits the audit trail for the root account, and enables all users with knowledge of the root password to log inn directly. Root access, besides from the system console, should only be allowed via sudo[13] or the su command. This requires the system administrators to leave an audit trail of the root access. Access to the su and sudo programs should de limited to the wheel group, where only the systems administrators are defined.

2.9.3 Separation of duties

Only two types of users have been defined on this server. These are:

- Systems administrators who maintains administer the server. These users belong to the wheel group.
- Normal users who accesses the server to transfer files using ftp. These users belong to the ftpusers group.

2.9.3 Least privilege

Only the systems administrators have a valid shell for logins. Knowledge of the root password is only given to this group. Normal users are only able to use ftp to access the server.

2.10 Backup policies and disaster preparedness

2.10.1 Backup policies

There are no written backup policies for the server ftpgw. The server is not backed up, and the current practice is to reinstall it in event of a serious crash. User accounts will be created manually after a reinstall, and new passwords reissued to the users. The reason given for this is that the server is only a gateway and data is automatically deleted after a week, and that there therefore is no need for backup since no data expect user accounts are stored permanently.

There should be a backup policy in place, which ensures a proper backup of the server. Although no data is stored on the server, a proper backup will decrease the time to get the server back up into production. A good backup means that user accounts don't need to be recreated or new passwords reissued. It also preserves changes made on the server to increase security and other configuration changes.

[13] Sudo can be found at: http://www.courtesan.com/sudo/

Some of these changes may be forgotten, or be difficult to recreate if one were to reconstruct them manually. Backups can also play an important role in reconstructing the timeline in the event of an attack. Comparing files between different backup sets and current files may give information of how and when an attacker compromised the system.

2.10.2 Disaster preparedness

There is no disaster preparedness plan for the server ftpgw. There exists documented plans for other systems, but as this one is not classified as critical, it is not included in these. This is not seen as a problem due to the server's role in the organisation. There should however be documented procedures concerning attacks and attempts to compromising the server. Smaller systems like this are often used as springboards in attacks on more critical and important servers (i.e. as sniffers on the network to gather login/password for other servers and to gather network topology information). It is therefore important that procedures be in place that insures that all attempts to compromise servers be centrally reported and investigated.

2.11 Other issues/vulnerabilities

2.11.1 File integrity checking

There is no form of integrity check performed on system files. Mechanisms for integrity checking of important system files should be implemented. An example of such tools is Tripwire[14]. Integrity checking monitors important files and alerts the system administrators in various ways if these files are changed. This can play an important role in discovering unwanted configuration changes and attempts at compromising the server. Integrity checking will also show exactly which important system files have been changed if an attacker is able to compromise a server. The system administrators are then able to patch the security hole, which let the attacker in, and restore the changed files. Without integrity checking the system will need to be completely reinstalled, since there will be no way of telling which files that have been modified.

Attackers often install rootkits and modify configuration files to hide activity, gain access to sensitive data or just to hinder normal operations. Rootkits typically contains common system binaries with backdoors, password sniffers or simply just modified system binaries like ps and ls that hide processes or files from being displayed. This will often go undetected if no form of integrity checking is done automatically on the server

[14] An Open Source version can be downloaded from: http://www.tripwire.org/downloads/index.php and a commercial version can be found at: http://www.tripwire.com/products/servers/index.cfm

3.0 Critical issues and vulnerabilities.

3.1 Switch from FTP to SCP/SFTP

FTP should be disabled, as it is an insecure protocol without strong encryption of data and login credentials.

All traffic between the client and server should be encrypted. This will have a number of benefits.

- man-in-the-middle attacks become difficult/impossible (unless he has access to the servers and clients private keys) since strong authentication is used.
- Data theft and manipulation becomes difficult/impossible since all data is encrypted in a strong and secure manner
- Sniffing of login/password combos becomes difficult/impossible since this information no longer is sent in clear-text, but is encrypted in a strong and secure manner.
- By encrypting all traffic, it will be difficult for an attacker to know which traffic is sensitive. He/she will need to record all traffic and try to decrypt everything in order to discover sensitive data. Since everything is securely encrypted, this will make the job close to impossible. This is at best extremely time consuming, especially since there is no way of knowing which data is relevant for him/her.

Ftp should therefore be disabled, and SSH used instead. SSH gives the same functionality through SCP and SFTP, and is secure. It uses strong encryption of both traffic and login credentials. SSH also uses host-keys to avoid man-in-the-middle-attacks and authenticate the server and client machines. The user will receive a warning if the server key changes from last connection. This is to alert the users that someone might try to impersonate the server in the other end, and to make the user check to make certain that the server is the correct one[15].

When SSH is used for file transfers instead of ftp it will require the users to have a valid shell in order for this to work. This might seem to imply that the users also will be able to log into the account and get shell access. This would negatively impact security, but it can be avoided. There exists a shell replacement called scponly, which can be used as a replacement for the normal shell. All normal users should have scponly as their shell. Some of the features of scponly (quoted from their web site[16]) are:

- *logging: scponly logs time, client IP, username, and the actual request to syslog*
- *chroot: scponly can chroot to the user's home directory, disallowing access to the rest of the filesystem.*
- *sftp compatibility. My testing of sftp against an scponly user worked great. This is probably the cleanest and most usable way for an scponly user to access files. (of course, sftp is not ssh1 compatible.)*
- *WinSCP 2.0 compatibility*
- *rsync compatibility as a compile time option*

[15] The key might change legitimately from time to time due to various reasons. The server administrator may for example have generated new host keys. A detected change in keys is therefore only given as a warning.

[16] scponly is found at: http://www.sublimation.org/scponly/

- *gFTP compatibility.*
- *security checks*

Scponly also has the ability to chroot to the users home directory as stated above. This further enhances security by disallowing the users access to other system files, which contains usable information[17] for an attacker.

3.2 Integrity checking of system configuration files and binaries

Integrity checking of system files and binaries should be implemented to offer better protection and warning, if the server should be compromised. The detailed explanation for this is given in chapter 2.11.1. These checks should be run daily, and checked against secure database. This database should either be on a separate protected server, or on read only media on the server itself (i.e. cdrom, write protected floppy etc). Anomalies should be reported both to the servers logs itself and a separate loghost.

3.3 SSH configuration

SSH needs to be properly configured to improve security. This requires that a set of parameters in the /etc/ssh/sshd_config file be configured. The parameters that need to be set and their values are:

Protocol 2
This disables SSH1 support. SSH1 has security issues due to protocol design deficiencies[18].

PermitRootLogin no
Root should never be able to log in anonymously. We should require the user to first log in to server using a normal account first. This improves security and helps provide an audit trail of root access.

PermitEmptyPasswords no
The use of passwords is one of the most basic types of authentication, and should not be bypassed.

3.4 Logging

The server should be configured to log events to a central server. This will make alarm filtering, monitoring and alarm correlation easier if done on a central server. It also makes it more difficult for an attacker to hide his/hers tracks, since logs on two separate servers needs to be manipulated. Further details can be found in chapter 2.3.2
The server ftpgw should be configured to log event to central server by adding this line to the file "/etc/syslog.conf":

```
*.crit;*.err;*.emerg;auth.warning                    @central.log.host.com
```

[17] Such information may include the password file, various configurations files, interface/routing information etc.

[18] As quoted from http://www.openssh.org/security.html *"OpenSSH has the SSH 1 protocol deficiency that might make an insertion attack difficult but possible. The CORE-SDI deattack mechanism is used to eliminate the common case. Ways of solving this problem are being investigated, since the SSH 1 protocol is not dead yet."*

3.5 Physical security

Physical security should be improved. The server is placed in an unlocked rack. This rack is located in a secure room, but multiple persons besides the servers' system administrators have access to the room. This implies the possibility for everyone with this access to also access the server physically. Physical access to a server will in most cases enable a person to become root on the server and bypass most normal security controls. More details can be found in chapter 2.8.1

The server rack should be locked, the keys kept secure and use of them should be logged in a secure manner. This should be practice for all the server racks. Since all servers in a single rack are operated and maintained by the same group of people, this is thought to be fairly easy to implement.

3.6 Server backup

There should be a backup policy in place. This should ensure a proper backup of the server. Although no data is stored on the server, a proper backup will decrease the time to get the server back up into production. A good backup means that user accounts don't need to be recreated or new passwords reissued. It also preserves changes made on the server to increase security and other configuration changes. Some of these changes may be forgotten, or be difficult to recreate if one were to reconstruct them manually. Backups can also play an important role in reconstructing the timeline in the event of an attack. Comparing files between different backup sets and current files may give information of how and when an attacker compromised the system.

3.7 Time synchronization using NTP

The server is not configured to use NTP. Time synchronisation should be used on all servers. By ensuring that all clocks are in sync, correlating and reconstructing events based on system logs will be much easier. Being able to pinpoint the exact time of attacks on servers can make a world of difference sometimes. Enable NTP by installing and configuring the software as detailed in chapter 2.3.3

3.8 Disable unneeded services and start scripts

Unneeded services and start scripts should be disabled. This reduces the possibilities for an attacker to compromise the server. The more services left running, the more possible services to exploit and hack. Atd, xinetd and FTP (if SFTP/SCP is used instead) should be disabled. This is detailed in chapter 2.1.1

3.9 Procedures for account revocation

Written procedures and policies for account revocation should be in place and enforced.

The leaders, which are responsible for ordering access servers, should also be responsible for ordering the removal of users when they no longer require access. HR should also be responsible for sending lists of all personnel that has left the organization in order to make sure that users are removed from all servers.

3.10 Remove unnecessary packages

The list of packages in Appendix C should be examined closely and unnecessary packages should be removed to improve security.

3.11 Tune TCP/IP parameters for security

Several TCP/IP parameters should be tuned for security. These parameters can be configured in the file /etc/sysctl.conf. The setting that needs to be set and their desired values are detailed in chapter 2.1.1 (negative: 8.11)

3.12 Disable coredumps

Coredumps should be disabled both to reduce the chance of DOS attacks and exposure of sensitive information. This is detailed in chapter 2.1.1 (negative: 8.11)

3.13 Disaster preparedness

Documented written procedures concerning attacks and attempts to compromise the server should be created and enforced. All attempts should be centrally reported, as other servers may be attacked also. There should also be documented procedures on how to reconstruct the server in case of disaster (fire, total server crash, flood etc).

3.14 Require password for single user mode

In order to require personnel booting into a single user root shell to provide the root password, an entry for /sbin/sulogin in run level S should be added to /etc/inittab. Adding the following line after the ”initdefault” entry does this:
“~~:S:wait:/sbin/sulogin”

3.15 Mounting filesystems with the NODEV and NOSUID options

Filesystems that doesn’t normally have devices in them should be mounted with the nodev option. This is achieved by having the parameter “nodev” in the fourth column in /etc/fstab for the entries for /var, /ftproot, /boot, /mnt/cdrom and /mnt/floppy.
We should not honour the SUID bit on files from removable media either. Add the parameter “nosuid” to the fourth column in /etc/fstab for the entries for /mnt/cdrom and /mnt/floppy.

3.16 Disable ftp access for system-users

System-users should have locked accounts and there should never be any need to log in via ftp to the server as these. Ftp access to these accounts should therefore be denied. Denying ftp login to users is done by adding the username (on a separate line) to the file /etc/vsftpd.ftpusers. In this case the following users needs to be added to the file: mailnull, nscd, apache, rpcuser, gopher and rpc.

3.17 Restrict administrative access to cron and at

Only root should have this access on this server. Restrict this access with the following commands:

```
“rm -f /etc/at.deny /etc/cron.deny”
“echo root > /etc/cron.allow”
“echo root > /etc/at.allow”
“chown root:root /etc/cron.allow /etc/at.allow”
“chmod 400 /etc/cron.allow /etc/at.allow”
“chmod 400 /etc/crontab”.
```

3.18 Remove unnecessary world-writeable files

There is no need for world-writeable files on this server. To avoid potential abuse, remove the following two unnecessary files:
/var/www/html/signatur.log
/var/www/html/guestbook.html

3.19 Written policies for storage of sensitive data

Written policies should be in place, which ensures that sensitive data is deleted from the server immediately after a successful transfer. This will reduce the potential exposure time on the host if it is compromised (except in the case where and attacker automatically copies data entering the server). The policies should also include a definition of sensitive data.

3.20 Security patch installation and management

There should be defined procedures to inform all system administrators when new patches become available, and when relevant security holes are found. This procedures should ensure that such patches be installed and documented. See details in chapter 2.2.

4.0 References

4.1 Written material

- The Center for Internet Security, LinuxBenchmark.pdf, URL: http://www.cisecurity.org/bench_linux.html
- OpenSSH's security page, URL: http://www.openssh.org/security.html
- Linux Firewall-related /proc Entries, URL: http://www.securityfocus.com/infocus/1711
- tcp(7) - Linux man page, URL: http://www.die.net/doc/linux/man/man7/tcp.7.html
- CERT® Advisory CA-2003-24 Buffer Management Vulnerability in OpenSSH, URL: http://www.cert.org/advisories/CA-2003-24.html
- Advisory RHSA-2003:279-17, URL: http://rhn.redhat.com/errata/RHSA-2003-279.html
- How scponly works, URL: http://www.sublimation.org/scponly/#howitworks
- Dhanjani, Nitesh. Hacknotes: Linux and Unix Security – Portable Reference, McGraw-Hill/Osborne, 2003
- Frisch, Aeleen. Essential System Administration, O'Reilly & Associates, 3rd Edition, 2002

4.2 Tools

- scponly, URL: http://www.sublimation.org/scponly
- CISscan for Linux, URL: http://www.cisecurity.org/bench_linux.html
- Nessus, URL: http://www.nessus.org/intro.html
- John The Ripper, URL: http://www.openwall.com/john/
- OpenSSH, URL: http://www.openssh.org
- Sudo, URL: http://www.courtesan.com/sudo/
- Tripwire,
  Open Source URL: http://www.tripwire.org/downloads/index.php
  Commercial version URL : http://www.tripwire.com/products/servers/index.cfm

4.3 Interviews

An interview was conducted with the system administrator, who is responsible for the server. This provided valuable information in the audit. The system administrator wishes to remain anonymous for security reasons.

Appendix A – Full CISscan report

(Negative 8.3 has been truncated for readability. It reports 2 lines for all normal users on the server)

```
*** CIS Ruler Run ***
Starting at time 20031017-14:13:15

Positive: 1.1 System appears to have been patched within the last month.
Negative: 1.2 sshd_config parameter Protocol is not set.
Negative: 1.2 sshd_config parameter PermitRootLogin is not set.
Negative: 1.2 sshd_config parameter PermitEmptyPasswords is not set.
Negative: 1.2 ssh_config must have 'Protocol 2' underneath Host *.
Positive: 2.1 inetd/xinetd is not listening on any of the miscellaneous ports checked in this item.
Positive: 2.2 telnet is deactivated.
Negative: 2.3 ftp not deactivated.
Positive: 2.4 rsh, rcp and rlogin are deactivated.
Positive: 2.5 tftp is deactivated.
Positive: 2.6 imap is deactivated.
Positive: 2.7 POP server is deactivated.
Positive: 3.1 Found a good daemon umask of 022 in /etc/rc.d/init.d/functions.
Negative: 3.2 xinetd is still active.
Positive: 3.3 Mail daemon is not listening on TCP 25.
Positive: 3.4 Graphical login is deactivated.
Positive: 3.5 X Font Server (xfs) script has been deactivated
Positive: 3.6 apmd is deactivated.
Positive: 3.7 Windows compatibility servers (samba) have been deactivated.
Positive: 3.8 NFS Server script nfs is deactivated.
Negative: 3.9 NFS script autofs not deactivated.
Positive: 3.10 NIS Client processes are deactivated.
Positive: 3.11 NIS Server processes are deactivated.
Positive: 3.12 RPC rc-script has been deactivated.
Positive: 3.13 netfs rc script is deactivated.
Positive: 3.14 printing daemon is deactivated.
Positive: 3.15 Web server is deactivated.
Positive: 3.16 SNMP daemon is deactivated.
Positive: 3.17 DNS server is deactivated.
Positive: 3.18 SQL database server is deactivated.
Positive: 3.19 Webmin GUI-based system administration daemon deactivated.
Positive: 3.20 Squid web cache daemon deactivated.
Negative: 3.21 Kudzu hardware detection program has not been deactivated.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/secure_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_redirects should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/eth0/accept_source_route should be set to 0.
Negative: 4.1 /proc/sys/net/ipv4/conf/lo/accept_source_route should be set to 0.
```

```
Negative: 4.1 /proc/sys/net/ipv4/tcp_max_syn_backlog should be at least 4096 to
handle SYN floods.
Negative: 4.2 /proc/sys/net/ipv4/conf/eth0/send_redirects should be set to 0.
Negative: 4.2 /proc/sys/net/ipv4/conf/lo/send_redirects should be set to 0.
Positive: 5.1 syslog captures authpriv messages.
Negative: 5.2 /etc/vsftpd.conf should have log_ftp_protocol set to yes.
Negative: 5.2 /etc/vsftpd.conf should not have xferlog_std_format set to yes.
Positive: 5.3 All logfile permissions and owners match benchmark recommendations.
Negative: 6.1 /var is not mounted nodev.
Negative: 6.1 /ftproot is not mounted nodev.
Negative: 6.1 /boot is not mounted nodev.
Negative: 6.2 Removable filesystem /mnt/cdrom is not mounted nosuid.
Negative: 6.2 Removable filesystem /mnt/cdrom is not mounted nodev.
Negative: 6.2 Removable filesystem /mnt/floppy is not mounted nosuid.
Negative: 6.2 Removable filesystem /mnt/floppy is not mounted nodev.
Negative: 6.3 PAM allows users to mount removable media: <floppy>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <cdrom>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <pilot>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <jaz>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <zip>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <ls120>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <camera>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <memstick>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <flash>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <diskonkey>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rem_ide>.
(/etc/security/console.perms)
Negative: 6.3 PAM allows users to mount removable media: <rio500>.
(/etc/security/console.perms)
Positive: 6.4 password and group files have right permissions and owners.
Positive: 6.5 all temporary directories have sticky bits set.
Positive: 7.1 rhosts authentication totally deactivated in PAM.
Positive: 7.2 /etc/hosts.equiv and root's .rhosts/.shosts files either don't exist, are
zero size or are links to /dev/null.
Negative: 7.3 User mailnull is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User nscd is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User apache is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User rpcuser is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User gopher is not present in /etc/vsftpd.ftpusers
Negative: 7.3 User rpc is not present in /etc/vsftpd.ftpusers
```

```
Positive: 7.4 X11 Server is blocked from listening on TCP port 6000.
Negative: 7.5 Couldn't open cron.allow
Negative: 7.5 Couldn't open at.allow
Negative: 7.6 The permissions on /etc/crontab are not sufficiently restrictive.
Negative: 7.7 No Authorized Only banner for vsftpd in file /etc/vsftpd.conf.
Negative: 7.8 xinetd either requires global 'only-from' statement or one for each
service.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: vc/11.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty7.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty8.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty9.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty10.
Negative: 7.9 /etc/securetty has a non console or tty 1-6 line: tty11.
Positive: 7.10 GRUB is password-protected.
Positive: 7.10 GRUB is password-protected.
Negative: 7.11 /etc/inittab needs a /sbin/sulogin line for single user mode.
Positive: 7.12 /etc/exports is empty or doesn't exist, so it doesn't need to be tuned for
privports.
Negative: 8.1 bin has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 daemon has a valid shell of /sbin/nologin.  Remember, the
/sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use
FTP.
Negative: 8.1 adm has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 lp has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell,
when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 mail has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 news has a valid shell of /bin/sh.  Remember, an empty shell field in
/etc/passwd signifies /bin/sh.
Negative: 8.1 uucp has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 operator has a valid shell of /sbin/nologin.  Remember, the
/sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use
FTP.
Negative: 8.1 games has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 gopher has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 ftp has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nobody has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpc has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin
shell, when found in /etc/shells, leaves a user potentially able to use FTP.
```

```
Negative: 8.1 vcsa has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 nscd has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 sshd has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpm has a valid shell of /bin/bash.
Negative: 8.1 mailnull has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 smmsp has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 rpcuser has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 pcap has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Negative: 8.1 apache has a valid shell of /sbin/nologin.  Remember, the /sbin/nologin shell, when found in /etc/shells, leaves a user potentially able to use FTP.
Positive: 8.2 All users have passwords
Negative: 8.3 User a34432 should have a minimum password life of at least 7 days.
Negative: 8.3 User a34432 should have a maximum password life of between 1 and 90 days.
Negative: 8.3 User a12566 should have a minimum password life of at least 7 days.
Negative: 8.3 User a12566 should have a maximum password life of between 1 and 90 days.
90 days.
Negative: 8.3 /etc/login.defs value PASS_MAX_DAYS = 99999, but should not exceed 90.
Negative: 8.3 /etc/login.defs value PASS_MIN_DAYS = 0, but should not be less than 7.
Negative: 8.3 /etc/login.defs value PASS_MIN_LEN = 5, but should be at least 6.
Positive: 8.4 There were no +: entries in passwd, shadow or group maps.
Positive: 8.5 Only one UID 0 account AND it is named root.
Positive: 8.6 root's PATH is clean of group/world writable directories or the current-directory link.
Positive: 8.7 No user's home directory is world or group writable.
Positive: 8.8 No group or world-writable dotfiles in user home directories!
Positive: 8.9 No user has a .netrc file.
Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/profile is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.login is 000 -- it should be stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/bashrc is 022 -- it should be stronger to block group-read/write/execute.
```

```
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be
stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /etc/csh.cshrc is 002 -- it should be
stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be
stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bash_profile is 000 -- it should be
stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be
stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.bashrc is 000 -- it should be
stronger to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger
to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.cshrc is 000 -- it should be stronger
to block group-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be
stronger to block world-read/write/execute.
Negative: 8.10 Current umask setting in file /root/.tcshrc is 000 -- it should be
stronger to block group-read/write/execute.
Negative: 8.11 Coredumps aren't deactivated.
Preliminary rating given at time: Fri Oct 17 14:13:16 2003

        Preliminary rating = 6.15 / 10.00

Negative: 6.6 Non-standard world-writable file: /var/www/html/signatur.log
Negative: 6.6 Non-standard world-writable file: /var/www/html/guestbook.html
Positive: 6.7 No non-standard SUID/SGID programs found.
Ending run at time: Fri Oct 17 14:13:16 2003

        Final rating = 6.31 / 10.00
```

Appendix B – Full NESSUS report

```
NESSUS SECURITY SCAN REPORT

Created 20.10.2003          Sorted by host names

Session Name : ftpgw
Start Time   : 20.10.2003 14:41:11
Finish Time  : 20.10.2003 14:41:34
Elapsed Time : 0 day(s) 00:00:22


Total security holes found : 11
             high severity : 1
              low severity : 8
             informational : 2


Scanned hosts:

Name                        High  Low   Info
---------------------------------------------
127.0.0.1                     1     8     2


Host: 127.0.0.1

Open ports:

   ssh (22/tcp)
   ftp (21/tcp)


Service: ssh (22/tcp)
Severity: High


You are running a version of OpenSSH which is older than 3.7.1

Versions older than 3.7.1 are vulnerable to a flaw in the buffer management
functions which might allow an attacker to execute arbitrary commands on this
host.

An exploit for this issue is rumored to exist.


Note that several distributions patched this hole without changing
the version number of OpenSSH. Since Nessus solely relied on the
```

```
banner of the remote SSH server to perform this check, this might
be a false positive.

If you are running a RedHat host, make sure that the command :
        rpm -q openssh-server

Returns :
 openssh-server-3.1p1-13 (RedHat 7.x)
 openssh-server-3.4p1-7  (RedHat 8.0)
 openssh-server-3.5p1-11 (RedHat 9)

Solution : Upgrade to OpenSSH 3.7.1
See also : http://marc.theaimsgroup.com/?l=openbsd-
misc&m=106375452423794&w=2
   http://marc.theaimsgroup.com/?l=openbsd-misc&m=106375456923804&w=2
Risk factor : High
CVE : CAN-2003-0693, CAN-2003-0695
BID : 8628


Service: ftp (21/tcp)
Severity: Low

Remote FTP server banner :
220 ready, dude (vsFTPd 1.1.0: beat me, break me)



Service: ftp (21/tcp)
Severity: Low

An FTP server is running on this port.
Here is its banner :
220 ready, dude (vsFTPd 1.1.0: beat me, break me)



Service: ssh (22/tcp)
Severity: Low


You are running OpenSSH-portable 3.6.1p1 or older.

If PAM support is enabled, an attacker may use a flaw in this version
to determine the existence or a given login name by comparing the times
the remote sshd daemon takes to refuse a bad password for a non-existent
login compared to the time it takes to refuse a bad password for a
valid login.

An attacker may use this flaw to set up  a brute force attack against
```

```
the remote host.

*** Nessus did not check whether the remote SSH daemon is actually
*** using PAM or not, so this might be a false positive

Solution : Upgrade to OpenSSH-portable 3.6.1p2 or newer
Risk Factor : Low
CVE : CAN-2003-0190
BID : 7482, 7467, 7342


Service: ssh (22/tcp)
Severity: Low

The remote SSH daemon supports the following versions of the
SSH protocol :

  . 1.33
  . 1.5
  . 1.99
  . 2.0



Service: ssh (22/tcp)
Severity: Low


The remote SSH daemon supports connections made
using the version 1.33 and/or 1.5 of the SSH protocol.

These protocols are not completely cryptographically
safe so they should not be used.

Solution :
 If you use OpenSSH, set the option 'Protocol' to '2'
 If you use SSH.com's set the option 'Ssh1Compatibility' to 'no'

Risk factor : Low


Service: ssh (22/tcp)
Severity: Low


You are running OpenSSH-portable 3.6.1 or older.

There is a flaw in this version which may allow an attacker to
bypass the access controls set by the administrator of this server.
```

```
OpenSSH features a mechanism, which can restrict the list of
hosts a given user can log from by specifying a pattern
in the user key file (ie: *.mynetwork.com would let a user
connect only from the local network).

However there is a flaw in the way OpenSSH does reverse DNS lookups.
If an attacker configures his DNS server to send a numeric IP address
when a reverse lookup is performed, he may be able to circumvent
this mechanism.

Solution : Upgrade to OpenSSH 3.6.2 when it comes out
Risk Factor : Low
CVE : CAN-2003-0386
BID : 7831


Service: ssh (22/tcp)
Severity: Low

An ssh server is running on this port


Service: ssh (22/tcp)
Severity: Low

Remote SSH version : SSH-1.99-OpenSSH_3.4p1
```

Appendix C – Complete listing of installed packages

(from the rpm database, using “rpm –qa” command)

```
gnome-mime-data-2.0.0-9
redhat-menus-0.26-1
filesystem-2.1.6-5
glibc-2.2.93-5
bzip2-libs-1.0.2-5
cracklib-2.7-18
e2fsprogs-1.27-9
gdbm-1.8.0-18
glib2-2.0.6-2
gmp-4.1-4
libart_lgpl-2.3.10-1
libacl-2.0.11-2
libgcc-3.2-7
libogg-1.0-1
libvorbis-1.0-1
mingetty-1.00-3
net-tools-1.60-7
perl-5.8.0-55
popt-1.7-1.06
slang-1.4.5-11
termcap-11.0.1-13
bash-2.05b-5
esound-0.2.28-1
fontconfig-2.0-3
ncurses-5.2-28
diffutils-2.8.1-3
grep-2.5.1-4
openssl-0.9.6b-29
procps-2.0.7-25
readline-4.3-3
sed-3.02-13
textutils-2.0.21-5
utempter-0.5.2-10
words-2-20
pam-0.75-40
sh-utils-2.0.12-3
modutils-2.4.18-2
initscripts-6.95-1
cyrus-sasl-md5-2.1.7-2
libuser-0.51.1-2
XFree86-libs-4.2.0-72
XFree86-Mesa-libGL-4.2.0-72
pango-1.1.1-1
zlib-1.1.4-4
libtiff-3.5.7-7
libxml-1.8.17-5
bonobo-activation-1.0.3-2
libbonobo-2.0.0-4
libgnomecanvas-2.0.2-1
python-2.2.1-17
util-linux-2.11r-10
fam-2.6.8-4
libgnome-2.0.2-5
libgnomeui-2.0.3-3
apmd-3.0.2-12
at-3.1.8-31
bc-1.06-10
bzip2-1.0.2-5
crontabs-1.10-4
dev-3.3.1-2
dialog-0.9b-20020519.1
dosfstools-2.8-3
eject-2.0.12-7
fbset-2.1-11
findutils-4.1.7-7
ftp-0.17-15
gpm-1.19.3-23
grub-0.92-7
hesiod-3.0.2-21
htmlview-2.0.0-6
isdn4k-utils-3.1-58
kbd-1.06-26
krbafs-1.1.1-6
kudzu-0.99.69-1
gzip-1.3.3-5
lha-1.14i-7
libpng10-1.0.13-5
libungif-4.1.0-13
libwvstreams-3.70-5
logrotate-3.6.5-2
lrzsz-0.12.20-14
lvm-1.0.3-9
mailx-8.1.1-26
make-3.79.1-14
man-1.5j-11
minicom-2.00.0-6
mt-st-0.7-6
mtr-0.49-7
nscd-2.2.93-5
ntsysv-1.3.6-3
```

```
ORBit-0.5.13-5
pam_krb5-1.56-1
parted-1.4.24-6
pax-3.0-4
ppp-2.4.1-7
pspell-0.12.2-14
pyOpenSSL-0.5.0.91-1
raidtools-1.00.2-3.3
rdist-6.1.5-24
reiserfs-utils-3.6.2-2
rmt-0.4b28-4
rootfiles-7.2-4
net-snmp-5.0.1-6
rpm-python-4.1-1.06
setserial-2.17-9
slocate-2.6-4
star-1.5a04-1
stunnel-3.22-4
syslinux-1.75-3
mkinitrd-3.4.28-1
iptables-1.2.6a-2
lilo-21.4.4-20
mkbootdisk-1.4.8-1
pciutils-2.1.10-2
rp-pppoe-3.4-7
tcpdump-3.6.3-3
telnet-0.17-23
timeconfig-3.2.9-1
traceroute-1.4a12-6
unzip-5.50-5
usbutils-0.9-7
kernel-pcmcia-cs-3.1.31-9
vim-minimal-6.1-14
anacron-2.3-23
whois-1.0.10-4
wvdial-1.53-7
zip-2.3-14
curl-devel-7.9.8-1
db4-devel-4.0.14-14
expat-devel-1.95.4-1
gmp-devel-4.1-4
hesiod-devel-3.0.2-21
krbafs-devel-1.1.1-6
libogg-devel-1.0-1
libusb-0.1.6-1
libuser-devel-0.51.1-2
libxml-devel-1.8.17-5
modutils-devel-2.4.18-2
openldap-devel-2.0.25-1
pam-devel-0.75-40
kudzu-devel-0.99.69-1
rpm-devel-4.1-1.06
newt-devel-0.51.0-1
zlib-devel-1.1.4-4
automake-1.6.3-1
automake15-1.5-4
bison-1.35-4
cdecl-2.5-25
cvs-1.11.2-5
diffstat-1.28-4
flex-2.5.4a-26
glibc-kernheaders-2.4-7.20
gcc-3.2-7
libf2c-3.2-7
libgcj-3.2-7
libgcj-devel-3.2-7
libgnat-3.2-7
libstdc++-devel-3.2-7
ltrace-0.3.10-12
autoconf-2.53-8
memprof-0.5.0-2
patchutils-0.2.14-3
pkgconfig-0.12.0-3
rcs-5.7-18
redhat-rpm-config-8.0-1
strace-4.4-8
vsftpd-1.1.0-1
comps-8.0-0.20020910
httpd-devel-2.0.40-8
openssh-3.4p1-7
openssh-server-3.4p1-7
glibc-common-2.2.93-5
hwdata-0.47-1
setup-2.5.20-1
basesystem-8.0-1
bdflush-1.5-21
chkconfig-1.3.6-3
db4-4.0.14-14
expat-1.95.4-1
glib-1.2.10-8
atk-1.0.3-1
iputils-20020124-8
libattr-2.0.8-3
libcap-1.10-12
libjpeg-6b-21
libstdc++-3.2-7
linc-0.5.2-2
mktemp-1.5-16
pcre-3.9-5
perl-Filter-1.28-9
```

```
shadow-utils-20000902-12
newt-0.51.0-1
libtermcap-2.0.8-31
audiofile-0.2.3-3
freetype-2.1.2-7
iproute-2.4.7-5
info-4.2-5
gawk-3.1.1-4
fileutils-4.1.9-11
ORBit2-2.4.1-1
psmisc-20.2-6
redhat-release-8.0-8
sysklogd-1.4.1-10
mount-2.11r-10
which-2.14-1
cracklib-dicts-2.7-18
authconfig-4.2.12-3
krb5-libs-1.2.5-6
SysVinit-2.84-5
cyrus-sasl-2.1.7-2
openldap-2.0.25-1
usermode-1.63-1
gtk+-1.2.10-22
Xft-2.0-1
xinetd-2.3.7-2
libpng-1.2.2-6
gtk2-2.0.6-8
libxml2-2.4.23-1
GConf2-1.2.1-3
libglade2-2.0.0-2
libxslt-1.0.19-1
rhpl-0.51-1
portmap-4.0-46
gnome-vfs2-2.0.2-5
libbonoboui-2.0.1-2
acl-2.0.11-2
ash-0.3.8-5
attr-2.0.8-3
bind-utils-9.2.1-9
cpio-2.4.2-28
cyrus-sasl-plain-2.1.7-2
dhclient-3.0pl1-9
dos2unix-3.1-12
ed-0.2-28
ethtool-1.6-2
file-3.37-8
finger-0.17-14
gnupg-1.0.7-6
groff-1.18-6
hdparm-5.2-1
autofs-3.1.7-33
irda-utils-0.9.14-6
jfsutils-1.0.17-3
kbdconfig-1.9.16-1
ksymoops-2.4.5-1
less-358-28
lftp-2.5.2-5
libelf-0.8.2-2
libtool-libs-1.4.2-12
imlib-1.9.13-9
lockdev-1.0.0-20
losetup-2.11r-10
lsof-4.63-2
mailcap-2.1.12-1
logwatch-2.6-8
MAKEDEV-3.3.1-2
man-pages-1.53-1
mouseconfig-4.26-1
mtools-3.9.8-5
netconfig-0.8.12-3
nss_ldap-198-3
gnome-libs-1.4.1.2.90-22
pam_smb-1.1.6-5
passwd-0.67-3
pinfo-0.6.4-7
procmail-3.22-7
aspell-0.33.7.1-16
python-optik-1.3-2
rdate-1.2-5
redhat-logos-1.1.6-2
rhnlib-1.0-1
dump-0.4b28-4
rpm-4.1-1.06
net-snmp-utils-5.0.1-6
setuptool-1.10-1
specspo-8.0-3
statserial-1.1-30
sudo-1.6.6-1
tar-1.13.25-8
kernel-smp-2.4.18-14
kernel-2.4.18-14
lokkit-0.50-18
nfs-utils-1.0.1-2
quota-3.06-5
tcp_wrappers-7.6-23
tcsh-6.12-2
time-1.7-19
tmpwatch-2.8.4-3
unix2dos-2.2-17
up2date-3.0.7-1
```

```
hotplug-2002_04_01-13
vim-common-6.1-14
vixie-cron-3.0.1-69
wget-1.8.2-3
wireless-tools-25-1
ypbind-1.11-2
yp-tools-2.7-3
curl-7.9.8-1
cyrus-sasl-devel-2.1.7-2
db4-utils-4.0.14-14
gdbm-devel-1.8.0-18
gpm-devel-1.19.3-23
krb5-devel-1.2.5-6
libcap-devel-1.10-12
libtermcap-devel-2.0.8-31
libusb-devel-0.1.6-1
libvorbis-devel-1.0-1
lockdev-devel-1.0.0-20
ncurses-devel-5.2-28
openssl-devel-0.9.6b-29
pciutils-devel-2.1.10-2
readline-devel-4.3-3
slang-devel-1.4.5-11
swig-1.1p5-20
libxml2-devel-2.4.23-1
automake14-1.4p6-3
binutils-2.13.90.0.2-2
byacc-1.9-22
cpp-3.2-7
dev86-0.16.3-4
doxygen-1.2.14-8
gdb-5.2.1-4
glibc-devel-2.2.93-5
indent-2.2.8-3
gcc-g77-3.2-7
gettext-0.11.4-3
gcc-java-3.2-7
gcc-gnat-3.2-7
gcc-c++-3.2-7
m4-1.4.1-11
libtool-1.4.2-12
patch-2.5.4-14
perl-CPAN-1.61-55
python-devel-2.2.1-17
rpm-build-4.1-1.06
splint-3.0.1.6-3
texinfo-4.2-5
anonftp-4.0-12
httpd-2.0.40-8
httpd-manual-2.0.40-8
openssh-clients-3.4p1-7
sendmail-8.12.8-9.80
```