Global Information Assurance Certification Paper

Consultant's Report from Auditing Linux

Jonathan Isner
August 3, 2004

Abstract

This paper consists of a report of a security audit conducted on the Linux operating system. The report details the description of the system and its data, the methodology used to perform the security test, a description of each test conducted, the result of each test and the associated risk of the setting being tested and the importance of the setting within the security framework. Also, recommendations to securely fix each finding are included in the report. This report encompasses a means to securely configure the Linux operating system, and can be considered a security guideline or baseline for securely operating the Linux operating system.

GIAC Enterprises

Linux File Server Configuration

Audit Report

Isner Security Inc,

August 2004

Table of Contents

GIAC Enterprises Sales Department
Linux File Server Configuration
Audit Results

1. Executive Summary

GIAC Enterprises procured Isner Security Inc. to conduct a security audit of GIAC's computer resources, specifically the Sales Department Linux ES Version 3 file server. This document details the methodology used and the results of the security audit of GIAC Enterprises' Sales Department File Server.

The security audit encompassed the execution of security tests directly on the Linux Server and the analysis of the subsequent results to assess whether the system configuration and controls meet the pre-stated security requirements from the GIAC Enterprises Security Plan dated May 28, 2004.

A total of **16** findings were discovered on the server out of 48 tests executed. After review and analysis of findings from the audit, it has been determined the overall technical security configuration of GIAC Enterprises' Sales Department File Server has an associated risk level of **Medium**. This determination is based on the vulnerabilities found during the audit, the related threat environment those vulnerabilities exist in, and the existing security controls in place.

1.1 Critical Issues and Recommendations

The following section is a brief description of the top ten vulnerabilities from the audit that are recommended for fixing. These ten findings are listed here because they pose the greatest threat to system security. Specific recommendations for fixing these findings are found in Section 4 of this report along with the complete results of the audit.

1.1.1 Outdated SNMP Version

Currently SNMP version 1 is running on the system. This is especially risky because it is an outdated version of the protocol that uses an unencrypted community string as the only authentication mechanism. An attacker could exploit the default community string to enumerate information on the structure of the network and its servers and devices. Attackers could use this information to plan attacks against the network.

1.1.2 Unnecessary SUID Files

Currently SUID scripts reside on the system, which are not necessary for system operation. SUID files allow the user who runs the file to assume the privileges of the file's owner. This situation is especially dangerous because it would give an attacker who is only able to compromise a regular user account could elevate their privileges on the system.

1.1.3 NIS Server

Currently, the NIS processes ypserv and ypasswd are running on the system. NIS is particularly risky because an attacker could use it to enumerate encrypted passwords, usernames, hostnames and associated IP addresses and mail aliases, thus giving the attacker an arsenal of information which to compromise the server.

1.1.4 Disaster Recovery Plan

Currently a Disaster Recovery Plan is not in place for the system. This is an extremely poor security practice. Without a Disaster Recovery Plan, in the event of a crisis, the system and its sensitive data may not be able to be recovered.

1.1.5 Superfluous Services

Currently there are many network services running on the system that are not needed for daily business operation. It is especially risky to run a lot of network services because every service running on a system is a potential security hole for an attacker to exploit. All network services carry both known and potential security flaws. Running some services have small risks, while running others put the system in great danger of being compromised.

1.1.6 R-Services

Currently rsh, rlogin and rcp are running on the system. These services allow users to run commands on remote machines, login to other machines and copy files between machines. These services are a frequent target of hacker exploits because their weak authentication schemes. The ability to connect to r-services would give an attacker unauthorized access to the system.

1.1.7 X Server

Currently the X server is running, which allows graphical logins. X-Windows is susceptible to buffer overflow attacks, denial-of-service attacks, and the ability for an attacker to control the display, in which they could capture keystrokes and read passwords.

1.1.8 TCP Wrappers

Currently the TCP Wrappers program is not used on the system. TCP Wrappers is an important tool because it provides access control to the network services on the system. Network services such as the services found in inetd are common targets of hackers because they allow hackers to connect to the server.

1.1.9 Root Account Logins

Currently the root account is allowed to login remotely. Most attacks occur from remote systems. The root account is the most targeted account on a UNIX system because of the elevated system privileges it has. Stopping remote logins directly to the root account will make it significantly more difficult for an attacker to compromise the root account.

1.1.10 Log Files Review

Currently a formal process is not in place for the review of system logs. This is a poor security practice and dangerous to system security because of the security information contained within the logs. The logs reveal potential attempts to compromise the system, as well as indicate attacks occurring on the system. If not reviewed regularly, this important security information would go unnoticed.

2. System Description

GIAC Enterprises' Sales Department employs ten Sales Associates, one Linux administrator and one security administrator. The Sales Department File Server houses sensitive customer information. This information includes account numbers, customer names and addresses, and financial information. This information is considered sensitive, and proprietary to individual customers.

The operating system used for the server is Red Hat Enterprise Linux version 3 based on the Linux kernel 2.4.21. The Linux operating system is hosted on the x86 Intel platform. The server is a Dell PowerEdge 700, 1024MB RAM, Intel Pentium 4 CPU 800MHz FSB 2.8Ghz, Intel Corp. 82547EI Gigabit Ethernet Controller, 1 80GB ATA hard drive, 48X IDE CD-RW/DVD ROM, and a ATI Technologies Inc Rage XL video adapter.

Logically the server resides on the GIAC Enterprises Local Area Network (LAN) behind a Cisco PIX firewall, which separates the server from the Internet. Currently this firewall is all that separates the server from the outside world, so a secure configuration of the Linux operating system is imperative.

The server is located in a physically secure server room within the GIAC Enterprises' Corporate Datacenter. The Physical Security Department is responsible for controlling security and access to the building. Access to the server room is controlled by electronic badge. Badge access to the server room is only given to employees with a need to access the room. All badge access is logged and reviewed by the Physical Security Department.

3. Audit Methodology

The Isner Security Inc. audit team consisted of two security engineers to conduct the audit. The Isner security engineers did not physically touch GIAC Enterprises' computer resources, GIAC Enterprises system administrators executed the commands while the Isner team witnessed and recorded the results. The entire audit was completed using manual commands run on the Linux operating system. The commands are detailed in Section 4 of this report. No automated tools or scanners were run on the system.

Test execution followed the stated test objectives defined in the GIAC Enterprises Linux File Server Configuration Audit Plan, dated June 11, 2004. The primary goal of the test objectives is to verify the existence and proper configuration of required security mechanisms provided by the Linux File Server Configuration. Therefore, test procedures developed to verify these objectives are technical in nature and required hands-on test execution.

The results of each test objective were ascertained using root access and executing commands in the shell.

The results of each test procedure are categorized into one of three ratings:

- **Pass** – The stated test objective is met.
- **Not Met** – The stated test objective is not met.
- **Not Tested** – The test objective is not applicable at the time of testing.

A risk level is assigned to each finding from the audit. The risk level assigned to each finding is factored by taking into account the likelihood of exploitation of the vulnerability and the impact to the system if the vulnerability is exploited. Risk is categorized into one of four ratings:

- **High** – There is a high likelihood the finding will be exploited.
- **Medium** – There is a medium likelihood the finding will be exploited.
- **Low** – There is a low likelihood the finding will be exploited.
- **Negligible** – There is no risk associated with the finding, the setting is secure.

All test results are documented in Section 4 of this report.

4. Test Results

4.1 Boot Loader Protection

An attacker can easily subvert the normal boot process on most default Linux boot loaders (LILO, GRUB). By password protecting the boot loader, the system

will boot normally, but will require a password if attempts are made to bypass LILO or GRUB. GIAC Enterprises uses LILO as the boot loader.

Test Executed:

1) # cat /etc/lilo.conf

Expected Output:

1) The following line is present in the file:

restricted
password=<*password*>

Test Result: Pass

Risk: An attacker with physical access to the system console could interrupt the system boot process and change the root password. This would give the attacker full access to system, while creating a denial-of-service for the actual system administrator or user. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None

4.2 Root Logins

Root shall only be allowed to log directly into the console, and only in emergency situations. The /etc/securetty file allows you to specify which TTY devices the root user is allowed to login to. Root access shall only be allowed after logging in with a regular user account and performing a 'su' to root. This method provides an audit trail of who is accessing the root account. Root logins can be prevented from SSH connections as well by ensuring the PermitRootLogin parameter in the sshd_config file is set to no.

Test Executed:

1) # cat /etc/securetty
2) # cat /etc/ssh/sshd_config

Expected Output:

1) All TTYs are commented out except for the console, tty1.

tty1
#tty2
#tty3

#tty4
#tty5
#tty6
#tty7
#tty8

2. PermitRootLogin no

Test Result: Not Met. The /etc/securetty file is not configured properly. The root account is able to login remotely.

Risk: Since the root account has such vast privileges on the system; the account should be well protected because it is the main target of attackers. If root is compromised, the entire system is compromised. Limiting the ability of the root account to logon is one way to help secure its use. If the root account were able to login remotely, accountability would be weakened. The risk of exploitation of this finding is **Medium.**

Recommendation: Disallow the root account to directly log in over the network by commenting out all the TTY entries except for the console in /etc/securetty.

4.3 Password Length

Passwords are the primary method for authenticating users to the system. Weak passwords are susceptible to many common attacks and password cracks. Passwords shall be at least eight characters in length, and contain a mixture of letters, numbers and special characters. The most common access method of breaking into a system through a network, over a modem connection or sitting in front of a terminal is through weak passwords[1].

Test Executed:

1) # cat /etc/login.defs

Expected Output:

1) PASS_MIN_LEN 8

Test Result: Pass

Risk: Passwords that are less than eight characters in length, and do not contain a mixture of characters could be cracked much easier than a password that conforms to a strong password policy. An attacker could use a password cracking tool such as John the Ripper on the password file to discover the passwords of all the accounts on the system, and use that account information to

[1] *Linux Unleashed, p. 846*

gain unauthorized entry to the system. This setting is configured correctly; the risk associated with setting is **Negligible.**

Recommendation: None

4.4 Password Aging

Password aging requires a user to change their password after a set amount of time. This practice makes password cracking more difficult because of periodic password changes.

Test Executed:

1) # cat /etc/login.defs

Expected Output:

1) PASS_MAX_DAYS 90
   PASS_MIN_DAYS 7
   PASS_WARN_AGE 7

Test Result: Not Met. Password aging is not configured (PASS_MAX_DAYS, PASS_MIN_DAYS).

Risk: If password aging is not employed, a user will not be required to periodically change their password. An attacker would have a much easier time cracking user account passwords if the passwords remain constant over time. If an attacker were able to successfully crack an account password, the attacker would have unauthorized system access. The risk of exploitation of this finding is **Low.**

Recommendation: In the /etc/logins.def file, set the PASS_MAX_DAYS parameter to 90 to force a password change every 90 days, and set the PASS_MIN_DAYS to 7 prevent passwords from being immediately changed to something different after a password change.

4.5 Temporary and Shared Accounts

Temporary, guest and test accounts should always be removed or disabled when they are not in use. Sharing accounts is a bad practice because it requires a system password be shared amongst multiple users.

Test Executed:

1) # grep test /etc/passwd
2) # grep guest /etc/passwd

3) # grep temp /etc/passwd

Scan the password file for any other intuitive account names that would indicate temporary or shared accounts.

Expected Output:

No output is produced because there are no test, guest or temporary accounts.

Test Result: Not Met. An account called "testuser" and an account called "guest" was found in the password file. Neither account was locked.

Risk: Test, guest or temporary accounts are often targets of attackers because they provide a level of anonymity since they cannot be traced to a specific system user. Also, when these types of accounts are created there is a tendency to assign them weak passwords that could be easily exploited by an attacker. The risk of exploitation of this finding is **Low.**

Recommendation: Remove the "testuser" and "guest" accounts from the system. If any test, guest or temporary accounts are required for day-to-day business operations, ensure that when they are not in use the accounts are locked. Also ensure the accounts have strong passwords assigned to them and only those users with a business need have access to the accounts. Test, guest and temporary accounts should only be used for the specific purpose they were created for.

4.6 Unnecessary Accounts

Linux systems typically have a number of default system accounts that are not associated with any actual users on the system. These accounts are installed on the system by default when the operating system is first installed. These accounts shall be locked and have their shell set to a null shell.

Test Executed:

1) # cat /etc/passwd

Scan the password file for system accounts.

Expected Output:

All system accounts are locked with the *LK* string in the password field. System accounts also have their shell set to a null shell, for example /dev/null.

Test Result: Pass

Risk: Every account on the system represents a potential door for an attacker to gain unauthorized entry to the system through. Every account that is installed on the system by default, but is not used should be locked or deleted from the system to close the door to the attacker. This setting is configured correctly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.7 Accounts with No Passwords

An account with a null password field means that any user on the system could log into the account without having to authenticate with a password. All accounts shall have strong passwords assigned to them or be locked.

Test Executed:

1) # cat /etc/shadow

Expected Output:

All accounts in the shadow file have passwords assigned to them.

Test Result: Pass

Risk: Accounts that do not require a password to authenticate to the system create an easy entry point to the system for an attacker, since the attacker would not have to take time cracking the password. This setting is configured correctly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.8 Accounts with UID 0

An account with the UID of 0 means the account has superuser privileges on the system. Only the root account shall have a UID of 0.

Test Executed:

1) # grep :0: /etc/passwd

Expected Output:

1) root

Test Result: Pass

Risk: Superuser privilege on the system should be very limited to only those accounts necessary to perform superuser functions. Having multiple superuser accounts on a system would give an attacker multiple avenues to gain unauthorized superuser access to the system. This setting is configured correctly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.9 Warning Banners

A warning banner is an important legal notice that warns both authorized and unauthorized users of the actions they take within the system. Warning banners are useful to aid in the prosecution of attackers. The /etc/issue file is the login banner displayed before the login prompt. The "/etc/issue.net" file is the login banner that users will see when they make a networked (i.e. telnet, SSH) connection to the system.

Test Executed:

1) # cat /etc/issue
2) # cat/etc/issue.net

Expected Output:

The text of the warning banner shown before successful logon to the system is displayed. The banner should identify the system is for authorized users only, and that improper use of the system will result in prosecution. The warning banner language should pertain to both authorized and unauthorized users, which would cover malicious insider users as well as attackers from outside.

Test Result: Not Met. A warning banner is not displayed before any logon to the system.

Risk: A warning banner provides legal notice to potential attackers that their actions are being audited, and that improper use of the system will result in prosecution. A warning banner often deters attackers from carrying out their attacks if they know their actions will be logged and the consequences will be severe. The risk of exploitation of this finding is **Low.**

Recommendation: Ensure the etc/issue and etc/issue.net files contain complete warning banners to authorized and unauthorized users for all console and network connections to the system. The warning banner should make known that the system is for authorized use only; all actions on the system are logged, and misuse of the system will results in legal consequences.

4.10 Session Timeout

A session timeout will close the current connection to the system once the system has been idle for a specified period of time. This helps prevent an attacker with physical access to the system console from exploiting an unattended system. The TMOUT variable is specified in seconds.

Test Executed:

1) # cat /etc/profile

Expected Output:

1) TMOUT=900
   export TMOUT

Test Result: Pass

Risk: Without a session timeout set, an attacker with physical access to the system could conduct various exploits to an unattended console with an active session logged into the system. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.11 Sticky Bit

If the sticky bit is set on a directory, an unprivileged user may not delete or rename files of other users in that directory, even if the user has write access to the directory. The only user who may delete or rename files in the directory would be the owner of the file.

Test Executed:

1) # find / -type d –perm 2 –exec ls –ld {} \;

Expected Output:

1) A list of world writable directories is displayed.

Verify “t” is the last letter in the permission string, indicating the sticky bit is set.

Test Result: Pass

Risk: The sticky bit ensures that unauthorized users are not able to rename or delete files that they do not own. This is especially important on world writable directories because this means anyone on the system can access the file, but

only the file owner should be able to delete the file or change its name. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.12 Unnecessary World Writable Files

Any user on the system can modify world writable files and directories. An attacker who gains access to the system could launch an attack by inserting malicious code or a Trojan horse in the world writable files. The attacker could also add or delete any files in world writable directories.

Test Executed:

1) # find / -type f \(-perm -2 -o -perm -20 \) -exec ls -lg {} \;
2) # find / -type d \(-perm -2 -o -perm -20 \) -exec ls -ldg {} \;

Expected Output:

A list of world writable files and directories is displayed. Review the files to ensure the necessity of the world writable permission.

Test Result: Pass

Risk: Any user on the system has the ability to modify world writable files and directories. This permission setting is particularly dangerous because any user on the system can modify these files by adding malicious code to them. Any user on the system could also delete any world writable file on the system. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.13 Unnecessary SETUID & SETGID Files

When an SUID or SGID program runs, the effective UID or GID becomes accelerated to that of the file owner rather than the user who is running the program. Therefore a user could have access to a file or program they normally would be restricted from accessing.

Test Executed:

1) # find / -type f \(-perm -04000 –o –perm –02000 \) –exec ls –ld {} \;

Expected Output:

1) All SUID and SGID files are listed.

If the program is SUID the execute (x) bit in the user permission will be changed to a 's'.

If the program is SGID the execute (x) bit in the group permission will be changed to a 's'.

The files shall be reviewed for necessity of SUID and SGID.

Test Result: Not Met. Several SUID scripts were found that the system administrator could not justify the need for.

Risk: SUID shell scripts owned by root could allow unauthorized users to obtain the highest privileges of the system. Group or world writeable permissions for SUID and SGID files would allow unauthorized access, resulting in modification or destruction of system files. The risk of exploitation of this finding is **Medium.**

Recommendation: Review the existing SUID and SGID files and ensure that permissions for the files are not group or world writeable. Also ensure that all SUID and SGID files are necessary for daily business operations.

4.14 Critical File Permissions

Test Executed:

1) # find /etc/init.d /etc/rc.d –type f –perm –0002 –ls
2) # ls –ld /etc /usr /usr/bin /usr/sbin
3) # ls –ld /root

Expected Output:

1) No startup files shall be world writable.
2) No files or directories in the /usr, /usr/bin, /usr/sbin directory are world writable.

3) No files or directories in the root home directory are world writable.

Test Result: Pass.

Risk: Attackers could take advantage of improperly assigned file and directory permissions. Startup files having world writeable permissions would allow an attacker to edit and install malicious code that will be executed at the time of the next system restart. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.15 Root UMASK

The umask is used to determine the file permission for newly created files. It specifies permissions that should not be given to the file by default when it is created.

Test Executed:

1) # umask

Expected Output:

2) The umask is set to 027, or a more restrictive value.

Test Result: Not met. The root umask is set is set to 022.

Risk: The root account should default to creating files that are secure. The current umask setting (022, rwxr-xr-x) allows unauthorized users to read and execute newly created files owned by root. If the umask is too liberal, new files would be created giving permission to users that would not need to access the files. The risk of exploitation of this finding is **Low.**

Recommendation: Set the root umask value to 027 (rwxr-x---) or 077 (rwx------).

4.16 R-Services

Rsh, rlogin and rcp allow users to run commands on remote machines, login to other machines and copy files between machines. These services are historically risky and a target of attackers because of their weak authentication. R-login does not necessitate the user type in a user name. The system receiving the r-login lets the user log in without providing a password.

Test Executed:

1) # chkconfig --list | grep rlogind
2) # chkconfig --list | grep rshd
3) # chkconfig --list | grep rcpd

Expected Output:

No output is returned because rlogin, rsh, and rcp services are disabled.

Test Result: Not met. All three daemons are active on the system.

Risk: The r-services are enabled, but according to the system administrator SSH is used in their place. Transmitting using r-services allows information to be sent in plain text, which permits data or keystrokes to be intercepted. Attackers can use packet sniffers in to discover usernames and passwords sent across the network in plain text. The risk of exploitation of this finding is **Medium.**

Recommendation: Since SSH is already in use on the system, the r-login, rsh and rcp services can safely be disabled without disrupting any system functionality. Disable the r-services.

4.17 Secure Shell (SSH)

The SSH suite of tools is a secure mechanism for carrying out remote operations on a server. SSH replaces the historically risky and oft exploited protocols telnet, rlogin and rsh, which transmit information across the network in plain text.

Test Executed:

1) # chkconfig | grep sshd

Expected Output: The SSH daemon is returned, SSH is in use.

Test Result: Pass

Risk: Without SSH, sensitive account and system information such as passwords would traverse the network in plain text. An attacker using a packet sniffer such as Ethereal could easily see the plain text information in transit and use it to gain unauthorized access to the system. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.18 Telnet

Telnet is a protocol used to log into remote systems over the network. It is insecure because it sends the username and password in clear text. Telnet gives the user a virtual terminal on the server. SSH shall be used instead.

Test Executed:
1) # chkconfig –list | grep telnetd

Expected Output:

1) Telnet is off for all runlevels.

Test Result: Pass.

Risk: Using telnet presents a great security risk to the system because it sends the account name and password over the network in plain text. An attacker using a packet sniffer such as Ethereal could easily see the plain text account name and password in transit and use it to gain unauthorized access to the system. An attacker could also hijack a telnet session using a technique called session hijacking, wherein the attacker would be able to execute whatever commands they desire after a legitimate user has logged in via telnet. This setting is configured correctly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.19 Risky Configuration Files

.rhosts files allows users to create a set of trusted hosts for using r-login. Trusted hosts in the .rhosts file would be able to use rlogin and rsh commands without a password. .shosts files are used to set up ssh without a password. While this is more secure than rsh, it is still not good security practice. .netrc files may contain unencrypted passwords for FTP. The .forward file is used to store addresses and programs for the user in Sendmail. .exrc files are startup files for the ex or vi editors[2].

Test Executed:

1) # find / -name .rhosts
2) # find / -name .shosts
3) # find / -name .netrc
4) # find / -name .exrc
5) # find / -name .forward

Expected Output:

No files should be returned. If files do exist, review their necessity.

Test Result: Pass.

Risk: .rhosts files are a frequent target of attackers who add their username to the .rhost file to gain remote shell access. An attacker could modify a .forward file to run a program of their choice when mail is received. An attacker could manipulate a .exrc file to create an SUID sh, which unauthorized full access of the system. These settings are configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

[2] *Practical UNIX and Internet Security, p.519*

4.20 X Windows

X-Windows is a network based window system with graphical display for UNIX systems. It is good for running remote jobs, but unfortunately the remote authentication scheme is weak, historically insecure and commonly exploited. Booting into runlevel 3, which is the normal text-based console, will disable the graphical login. The XFree86-xfs package is a font server that can also serve fonts to other X servers remotely. It should be disabled as well. X servers listen on TCP port 6000. X Windows uses an insecure authentication protocol, which an attacker could easily exploit and gain access to the system. The X server shall be prevented from listening on port 6000.

Test Executed:

1) # chkconfig --list | grep xfs
2) # netstat –a

Expected Output:

1) xfs is off for all runlevels.
2) The X Server is not listening on port 6000.

Test Result: Not Met. The xfs daemon is active and the X Server is listening on port 6000.

Risk: X-Windows is susceptible to buffer overflow attacks, denial-of-service attacks, and the ability for an attacker to control the display, in which they could capture keystrokes and read passwords. Despite the fact the system boots into run-level 3, the xfs daemon is active and the X Server is listening on port 6000. The risk of exploitation of this finding is **Medium.**

Recommendation: Disable the xfs daemon. Close TCP port 6000 to prevent it from listening for connections.

4.21 Kernel Network Parameters

Network related kernel parameters help harden the network stack against various attacks such as denial-of-service and spoofing. For example, log_martians logs packets with an impossible source address, which is an address that isn't part of the same network as the local machine. rp_filter helps protect against spoofing by dropping packets that aren't received on the normal interface.

Test Executed:

1) # cat /etc/sysctl.conf

Expected Output:

1) net.ipv4.ip_forward = 0
   net.ipv4.tcp_max_synbacklog = 4096
   net.ipv4.conf.all.log_martians = 1
   net.ipv4.conf.all.rp_filter = 1
   net.ipv4.conf.all.accept_source_route = 0
   net.ipv4.conf.all.send_redirects = 0
   net.ipv4.conf.all.accept_redirects = 0
   net.ipv4.conf.all.secure_redirects = 0

Test Result: Pass

Risk: Poorly configured IP options open the server up to various network-based attacks. A type of attack called IP Spoofing makes it possible for an IP address to be falsely assumed by an attacker, tricking a system to think the attacker's traffic is coming from a legitimate IP address. A man-in-the-middle attack occurs when someone between two communicating parties is monitoring the communication. "Man-in-the-middle attacks are like someone assuming your identity in order to read your message. The person on the other end might believe it is you because the attacker might be actively replying as you to keep the exchange going and gain more information.[3]" The purpose of a denial-of-service attack is to deny legitimate users access to system services. These settings are configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.21 System Resource Limits

The /etc/security/limits.conf file can be used to control and limit resources for the users on your system. It is important to set resource limits on all your users so they can't inadvertently perform denial of service attacks by running too many processes and exhausting memory resources.

A limit can also be set on the size of core files. Core files are generated when a program fails. They provide information on the problem. Limits should be set so no core files are generated on the system. However, core files may be useful to developers for debugging crashed programs, so they may be used on development systems.

Test Executed:

1) # cat /etc/security/limits.conf
2) # find / -local –name core –type f –perm

[3] *Microsoft TechNet (http://www.microsoft.com/windows2000/techinfo/reskit/en-us/default.asp?url=/windows2000/techinfo/reskit/en-us/cnet/cndb_ips_ddui.asp)*

Expected Output:

1) # Prevent core dumps globally
* hard core 0
* soft core 0

2) # Other per-user limits
* soft nproc 64
* hard nproc 128
* soft nofile 256
* hard nofile 1024

2) No core files are output.

Test Result: Not met, core files were found to exist on the system.

Risk: Core files can cause the file system to fill up. Core files are also world readable, and can contain sensitive information, which could be read by an attacker. The risk of exploitation of this finding is **Low.**

Recommendation: Instead of dumping core files, set the system to log the problem and exit.

4.22 File System Size

File system size shall be monitored to ensure that any partition does not reach capacity, thus causing a denial-of-service to legitimate users. File systems shall not exceed 85% capacity.

Test Executed:

1) # df –k

Expected Output:

1) No file system or partition exceeds 85% capacity.

Test Result: Pass

Risk: If a partition becomes full, the system may not be able to log new security events. Legitimate users would not be able to write to new files on the system, this creating a denial-of-service condition. This setting is configured correctly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.23 TCP Wrappers

By default Red Hat Linux allows all service requests to active network services on the system. The TCP Wrappers program provides access control to network services. Network services such as SSH, Telnet, and FTP, use TCP wrappers to stand guard between an incoming request and the requested service. TCP Wrappers is controlled from two files, etc/hosts.allow and etc/hosts.deny. The hosts.allow file specifies which IP addresses are granted access, and the hosts.deny file specifies which should be denied.

Test Executed:

1) # which tcpd
2) # cat /etc/hosts.allow
3) # cat /etc/hosts.deny

Expected Output:

1) # /usr/sbin/tcpd
2) Only valid IP addresses cleared to communicate with the system are listed.
3) A default deny statement exists (i.e., ALL:ALL:DENY)

Test Result: Not Met. TCP Wrappers is not used on the system.

Risk: TCP Wrappers provides very granular access to network services running on a Linux server. Allowing network services to every single computer on the Internet is a major security risk because it opens up the system to outside connections on each of the network services being run. The risk of exploitation of this finding is **Medium.**

Recommendation: Network services should be configured to server computers on the GIAC internal network, but the offerings should be restricted to computers outside of the network[4]. Implement TCP Wrappers to restrict network services on the system. A default deny statement shall exist in the hosts.deny file to deny any IP address not explicitly listed in the hosts.allow file.

4.24 FTP, TFTP

The File Transfer Protocol (FTP) is a risky service because it transmits data unencrypted, which allows attackers to sniff passwords and other data off the network. Trivial File Transfer Protocol (TFTP) allows users to transfer files to and from a remote machine without providing a password. It is normally used only for booting diskless workstations. Network devices such as routers and firewalls can use TFTP servers to read and write their configuration files. If the system is being used for FTP, the /etc/ftpusers file shall be used. /etc/ftpusers is a list of

[4] *Practical UNIX and Internet Security, p.484*

users who are NOT allowed to FTP into the system. The root account shall be in this file, as well as any unprivileged users who shouldn't be using FTP.

Test Executed:

1) # chkconfig --list | grep ftp
2) # cat /etc/ftpusers

Expected Output:

1) All FTP daemons (i.e., vsftpd, wu-ftpd, gssftpd) are off for all runlevels. The tftp daemon is off for all runlevels.

2) All system accounts (i.e., root, bin, uucp, sys, adm, lp, nuucp, listen, daemon, news, nobody, nobody4) are in the ftpusers file.

Test Result: Pass.

Risk: FTP is a risky service because it is unencrypted, which allows attackers to sniff passwords and other data off the network. The TFTP directory contains information that could help an attacker carry out an attack on the system. If routers and firewalls use a TFTP server, their configuration files would be stored in the TFTP directory. FTP and TFTP are disabled on the system; the risk associated with these settings is **Negligible.**

Recommendation: None.

4.25 Sendmail

Sendmail is a Mail Transport Agent (MTA) that uses the Simple Mail Transport Protocol (SMTP) to deliver email. Although many MTAs are capable of encrypting traffic between one another, most do not, which puts the email at risk being sent unencrypted. Only mail servers need to listen on port 25 for incoming email, so the Sendmail daemon can be disabled on non-mail server systems.

Test Executed:

1) # /usr/lib/sendmail –d0 –bt < /dev/null | grep i
2) # cat /etc/sysconfig/sendmail
3) # grep PrivacyOptions /etc/mail/sendmail.cf
4) # grep LogLevel /etc/mail/sendmail.cf
5) # ls –l /etc/mail/sendmail.cf

Expected Output:

1) The Sendmail version is at least 8.9.3.

2) DAEMON=no
3) PrivacyOptions=authwarnings, noexpn, novrfy, goaway, restrictqrun, restrictmailq
4) LogLevel=9
5) Permission setting is 644; owner is root.

Test Result: Pass.

Risk: Default installations enable the Sendmail mail server. There have been numerous vulnerabilities identified in the Sendmail program over the years. Most of the exploits are successful against older and unpatched versions of Sendmail. Current versions of Sendmail (versions after 8.9.3) address these known vulnerabilities. Since this server is not a mail server, Sendmail has been properly secured. These settings are configured correctly; the risk associated with these settings is **Negligible.**

Recommendation: None.

4.26 DNS

The Berkeley Internet Name Domain (BIND) package is the most generally accepted implementation of the DNS, which allows converts hostnames into the IP address. Unless the system is a name server, there is no need to run a DNS server on the system.

Test Executed:

1) # chkconfig --list | grep named

Expected Output:

Named is off for all runlevels

Test Result: Pass

Risk: DNS is susceptible to giving away sensitive information, buffer overflow cache poisoning attacks and denial-of-service attacks. DNS is disabled on the system; the risk associated with DNS is **Negligible.**

Recommendation: None.

4.27 NFS

The Network File System (NFS) is an RPC service used with portmap to share file systems over the network to other systems. Using NFS a workstation could access a remote file system as if it were a partition on their own system. If the

system is an NFS server and needs to export file systems using NFS service, the /etc/exports file must be configured with the most restrictive access possible. This means not using wildcards, not allowing root write access, and mounting read-only wherever possible.

Test Executed:

1) # chkconfig --list | grep nfs
2) # showmount –e

Expected Output:

1) NFS is off for all runlevels.
2) A list of exported file systems is shown. Export mounts are read only.

Test Result: Pass.

Risk: NFS passes all information unencrypted over the network. File systems can also be exported to clients allowing write access. This would enable the person who exported the file system to make changes to files on the NFS server. Another risk is the NFS server could export file systems that are not needed by the client, thus giving away system information that was not intended to share. NFS is disabled on the system; the risk associated with NFS is **Negligible.**

Recommendation: None.

4.28 NIS

Network Information Service (NIS) is used with the system portmapper as a distributed database that shares usernames, passwords, and other sensitive information to any computer claiming to be within its domain. ypbind runs on the client machine and broadcasts to find NIS servers. ypserv runs on the server and could be compromised by an attacker to send out bogus passwords known by the attacker.

Test Executed:

1) # chkconfig --list | grep yp

Expected Output:

1) Verify ypbind (client process), ypserv (server process) and yppasswd (server process) is off for all run levels.

Test Result: Not Met. Ypserv and ypasswd are running on the system.

Risk: NIS is a historically insecure protocol, making it risky to use. An attacker could trick an NIS server into thinking they are a legitimate client, thus disclosing encrypted passwords, usernames, hostnames and associated IP addresses and mail aliases to the attacker. The risk of exploitation of this finding is **High.**

Recommendation: Since NIS is not used on the system, the ypserv and ypasswd daemons can safely be disabled without disrupting any system functionality. Disable the ypserv and ypasswd daemons.

4.29 NTP

Network Time Protocol (NTP) synchronizes the system clock with a remote timeserver or time source (such as a satellite). This is very important in order for the system to maintain accurate timestamps on logs.

Test Executed:

1) # ntptrace

Expected Output:

1) server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp/drift
restrict default ignore
restrict 127.0.0.0 mask 255.0.0.0
authenticate yes

Test Result: Not met, time synchronization is not employed.

Risk: It's important to configure the system to synchronize its internal clock with an accurate time source. Inaccurate system time would weaken the analysis and evidence of an attack because the audit log timestamps would be incorrect.

Recommendation: Sync the server to an external time source using the /etc/ntp.conf file. The most basic ntp.conf file will simply list 2 servers, one that it wishes to synchronize with, and a pseudo IP address for itself (in this case 127.127.1.0). The pseudo IP is used in case of network problems or if the remote NTP server goes down. NTP will synchronize against itself until it can start synchronizing with the remote server again. It is recommended to list at least 2 remote servers that you can synchronize against. One will act as a primary server and the other as a backup[5].

[5] *The Linux Systems Administrator Guide*
http://www.tldp.org/LDP/sag/html/x2869.html

4.30 RPC

Remote Procedure Call (RPC) services are widely used for distributed network implementations. They allow for remote execution of commands. RPC services contain many flaws that are well documented and popular to exploit with the hacker community.

Test Executed:

1) # rpcinfo –p

Expected Output:

1) No output is produced; RPC services are not running. Some RPC services must be for services such as NFS. If any RPC services are listed, the administrator shall justify their existence.

Test Result: Pass.

Risk: Recently hackers have used insecure systems running RPC services to propagate major damaging worms over the Internet. RPC services are often exploited through buffer overflow attacks because RPC programs have weak error checking facilities. No RPC services were found running on the system; the risk associated with setting is **Negligible**.

Recommendation: None.

4.31 Portmapper

Portmapper starts the system portmapper that is used by all RPC based services (NIS, NFS) to assign ports on a server for clients wishing to communicate with the server. The portmapper service runs on TCP and UDP port 111.

Test Executed:

1) # chkconfig --list | grep portmap

Expected Output:

Portmap is off for all runlevels. If NFS is utilized, portmap is required.

Test Result: Pass.

Risk: The portmapper service must be running in order for RPC services to function. The portmapper service is not running on the system; the risk associated with setting is **Negligible**.

Recommendation: None.

4.32 SNMP

Simple Network Management Protocol (SNMP) is used for network monitoring and management of TCP/IP devices. Multiple vulnerabilities have been found in SNMP. Messages are exchanged between SNMP management stations and network devices that run the agent software. The method by which these messages are handled and the authentication mechanism behind such message handling both have significant exploitable vulnerabilities

Test Executed:

1) # chkconfig –list | grep snmpd
2) If SNMP is running, execute the following: # more /etc/snmp/snmp.conf

Expected Output:

1) SNMP is off for all runlevels.
2) The read-community value should not be set to a default or common community name, such as “public”. There should be no write-community value.

Test Result: Not Met. SNMP version 1 is running on the system. Public community string is used.

Risk: SNMP versions 1 and 2 use an unencrypted community string as the only authentication mechanism. The default community string used by most devices is “public,” which is well known to attackers, and rarely changed by administrators. Attackers can exploit the default community string to reconfigure or shut down devices remotely. Sniffed SNMP traffic can reveal information on the structure of the network and the systems and devices that make it up. Attackers could use this information to plan attacks against the network. Risk of this finding being exploited is **High.**

Recommendation: SNMP should be disabled if it is not necessary for a mission-critical function. If SNMP is necessary, upgrade to SNMP version 3. Change the community names to something difficult to guess.

4.33 System Services

Red Hat Linux systems have many unnecessary services enabled by default. The principle of least privilege should be applied to running services on a system. Only services necessary for a mission-critical business function should be actively running on a server. Unneeded services shall be disabled to close the security hole. If SSH is used it is possible to remove nearly all the xinetd services because of SSH’s secure login and file transfer mechanisms.

Test Executed:

1) # chkconfig --list

Expected Output:

1) A list of active services is displayed, as well as a list of xinetd services. Review the list for unnecessary boot services and unnecessary xinetd services.

Test Result: Not Met. The following services are running on the server: Finger, POP, NNTP, SNMP, rexec, rlogin, rsh and xfs.

Risk: Every service running on a system is a potential security hole for an attacker to exploit. All network services carry both known and potential security flaws. Running some services have small risks, while running others put the system in great danger of being compromised. The risk of exploitation of this finding is **Medium.**

Recommendation: Review the list of running services. Disable any service that is not needed to maintain a mission-critical business function on the server.

4.34 Cron

Cron is a daemon that allows recurring tasks to be scheduled and executed automatically according to the schedule. The at command is used to schedule a one-time task at a specific time. The cron.allow file lists users who are allowed to run the crontab command to modify cron jobs. Only root shall be in the cron.allow file. If the file cron.allow exists, only users listed in it are allowed to use cron, and the cron.deny file is ignored. If cron.allow does not exist, all users listed in cron.deny are not allowed to use cron. Any user not listed in the cron.deny file would be allowed to use cron. For production systems only the cron.allow file shall be used. All crontab files shall be owned by root and readable only by root.

Test Executed:

1) # find / -name cron.deny
2) # find / -name at.deny
3) # find / -name cron.allow
4) # find / -name at.allow
5) # more cron.allow (if cron is used)
6) # more at.allow (if at is used)
7) # ls –ld /var/spool/cron

Expected Output:

1) No output is produced.
2) No output is produced.
3) No output is produced.
4) No output is produced.
5) root
6) root
7) The owner is root and the permission is 400.

Test Result: Pass.

Risk: The cron.allow and at.allow files are a list of users who are allowed to run the crontab and at commands to submit jobs to be run at scheduled intervals. Only the system administrator needs the ability to schedule jobs. An attacker who gains unauthorized access to the system could manipulate the crontab and at commands to schedule tasks to occur on the system. For example the attacker could schedule the a system backup to a server or media of their choice, and have it run during off hours when the system is not being monitored. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.35 Security Updates

Vendors release updates and patches when they become aware of new security vulnerabilities in their software. It is critical to stay current with these releases, download and install the security patches. Security patches shall always be tested in a non-production environment before being installed on production systems. Red Hat releases errata to address bugs, provide enhancements, or to fix security vulnerabilities. The individual responsible for keeping the system up to date with security patches should subscribe to security alert mailing lists such as Bugtraq and the Red Hat Linux Security Mailing List. http://www.redhat.com/security contains security alert descriptions and downloads for all Red Hat systems.

Test Executed:

Ensure the system is up to date with the latest vendor security updates and patches. Ensure that a procedure is in place for testing patches before they are applied to production systems.

Expected Output:

The system has current security patches applied. A process is in place for identifying security patches, either a manual process where the administrator reviews the Red Hat security web site, or an automated process using the Red

Hat Network Update Module. All patches and fixes are thoroughly tested before being applied to the production server.

Test Result: Pass.

Risk: If a system is not patched with current vendor security patch releases, the system is at risk of being exploited by well-known, published vulnerabilities. This setting is configured properly; the system is up to date with current security patches, therefore risk associated with setting is **Negligible**.

Recommendation: None.

4.36 Anonymous Shutdowns

By default, /etc/inittab specifies that the system is set to shutdown and reboot in response to a [Ctrl]-[Alt]-[Del] key combination used at the console.

Test Executed:

1) cat /etc/inittab

Expected Output:

1) The following line is commented out:: # ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Test Result: Pass.

Risk: This setting leaves the system open to denial-of-service if an attacker issues the [Ctrl]-[Alt]-[Del] command at the console. An attacker could shutdown the system, effectively dropping all the legitimate user sessions connected to the server. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.37 Printing

If the system is not used to print files, or as a print server, there is no need to run the print daemons, which traditionally have poor security.

Test Executed:

1) # chkconfig --list | grep lpd
2) # chkconfig --list | grep cupsd

Expected Output:

1) The lpd daemon is off for all runlevels.
2) The cupsd daemon is off for all runlevels.

Test Result: Pass.

Risk: Active printing services represent another security hole waiting to be exploited by an attacker. Since this server's mission requires no printing functionality, no printer daemon or service is actively running on the system. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.38 Name Server Cache Daemon

Tuning the Name Server Cache Daemon (nscd) to an appropriate minimal level of caching can address many potential security issues. In particular, the passwd file, group file, network lookups from NIS and NFS and Role Based Access Control (RBAC) configuration information shall not be cached.

Test Executed:

1) # grep enable /etc/nscd.conf

Expected Output:

1) enable-cache passwd no
   enable-cache group no
   enable-cache exec-attr no
   enable-cache prof -attr no
   enable-cache user-attr no

Test Result: Pass.

Risk: The nscd will cache sensitive information such as the passwd file, the group file and role based access control information on the server. An attacker who can access the cache on the server could compromise this information. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.39 File System Mounting

The /etc/fstab file contains descriptive information about the various file systems. Only root shall be allowed to mount files and file systems. The nodev option prevents users from mounting unauthorized devices.

Test Executed:

1) # ls –l /etc/fstab
2) # more /etc/fstab

Expected Output:

1) The permission is set to 644.
2) The user option is not used in any fstab entries. The /usr file system is read only (ro). The rest of the file systems are nosuid, nodev. Removable media shall be mounted nosuid.

Test Result: Pass.

Risk: The system shall prevent users from mounting unauthorized devices. An attacker could mount a USB storage device, or CD-ROM drive to extract files from the system. Mounting file systems "nosuid" will prevent the introduction of SUID programs to the system, which could lead to privilege escalation of regular user accounts. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.40 Console Permissions

When logging in through the console, the etc/security/console.perms file governs the permissions given to users. By default, the user logged into the console is owns the floppy and CD-ROM drives.

Test Executed:

1) # cat /etc/security/console.perms

Expected Output:

1) All references to floppy, cdrom or any removable media device are removed or commented out of the file.

Test Result: Pass.

Risk: An attacker with physical access to the console could load malicious code onto the system via the floppy or CD-ROM drives without needing the root password. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.41 Syslog

Logging is an important part of security because it enables the administrator to keep track of all activity on the system. A great deal of security information can be recorded in the various system logs. It is also helpful to reconstruct events after a system is compromised. Syslog is a general logging facility configurable on the host. Any program can generate a syslog message containing the program name, the facility, priority, and the log message.

Test Executed:

1) # ps –ef | grep syslogd
2) # more /etc/syslog.conf

Expected Output:

1) The syslog daemon is active.
2) authpriv and auth logging is turned on and saved to /var/log/secure.

Test Result: Pass.

Risk: The risk of not using syslog would be the possibility of security events occurring on the system and going unnoticed. The original configuration file for syslog does not log AUTH messages to any files. AUTH messages should be logged to keep track of who logs into the system. If these messages are not logged, an attacker logging into the system via rogue accounts would go unnoticed. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.42 Lastlog

The lastlog displays the login name, port and last login time for all the accounts on the system.

Test Executed:

1) # lastlog

Expected Output:

1) A list of the accounts is displayed along with their last login. Review the list for any unauthorized logins.

Test Result: Pass.

Risk: The risk of not using the lastlog would be the possibility of unauthorized logons to the system going unnoticed. If a rogue account, or service account that is never logged into appears on the lastlog, it would be a sign that the system has been compromised. This setting is configured properly; the risk associated with setting is **Negligible**.

Recommendation: None.

4.43 Xinetd Logging

Xinetd is the replacement for inetd. Many of the risky protocols such as telnet, finger, rlogin, and rsh are a part of xinetd. The use of xinetd needs to be audited to ensure unauthorized users are not accessing it.

Test Executed:

1) # more/etc/xinetd.conf

Expected Output:

1) log_type = SYSLOG authpriv
   log_on_success = HOST PID DURATION USERID
   log_on_failure = HOST USERID ATTEMPT

Test Result: Pass.

Risk: Many of the risky protocols such as telnet, finger, rlogin, and rsh are a part of inetd. Incoming connections of inetd services shall be logged to syslog to ensure unauthorized users are not accessing the services within inetd. This setting is configured properly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.44 Log Protection

The log files shall only be readable and writable by root. Unprivileged users shall not be allowed to view or modify the log files. Attackers who can access the system with regular user privilege would not be able to erase the logs and cover their tracks.

Test Executed:

1) ls –al /var/log/secure

2) ls –al /var/log/boot.log
3) ls –al /var/log/cron
4) ls –al /var/log/maillog
5) ls –al /var/log/messages

Expected Output:

All files have a permission setting of 644 or more restrictive.

Test Result: Pass.

Risk: The risk of having too lenient of privileges on log files is that attacker would easily be able to cover their tracks after compromising the system by modifying the log files to delete entries or events created while they were hacking the system. This setting is configured properly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.45 Log Review

Audit logs become nothing more than an ineffective waste of disk space if they are not constantly reviewed for anomalies or suspicious activity. A system administrator shall review log files manually or via an automated process on a regular basis.

Test Executed:

1) Verify the process for reviewing logs.

Expected Output:

1) An administrator is responsible for reviewing the logs on a daily basis.

Test Result: Not Met. Currently there is no manual process in place to review the audit logs, and no automated tool is used either.

Risk: The system log files tell a great deal about the activity that takes place on the system. Log files hold information of possible upcoming attacks by showing traffic of someone probing the system. Log files hold information on attacks that are taking place by logging traffic and events as the attacker is accessing and modifying system files and data. Log files are an invaluable resource after an attack has occurred, to help reconstruct the attack itself and prevent future attacks. If the log files are not reviewed, potential attacks could go unnoticed and they would not be stopped from occurring, or attacks could be happening without the administrator's knowledge. The risk of exploitation of this finding is **Medium.**

Recommendation: An administrator should be assigned to review logs on a periodic basis. Manually reviewing logs can be time consuming and tedious. Swatch is a constant monitor of the log files, designed to alert the administrator of log activity based on certain triggers set up by the administrator. Swatch started out as the "simple watchdog" for actively monitoring log files produced by UNIX's syslog facility. It has since been evolving into a utility that can monitor just about any type of log[6]. Swatch can be found at http://swatch.sourceforge.net/.

4.46 Log Rotation

The purpose of rotation log files off the system is to preserve disk space, and preserve the actual log files. Most log files are located in the /var/log directory. Log files can grow to be quite large, consuming disk space and degrading system performance, potentially causing a denial of service. Log files shall be rotated off the system on a regular basis to preserve disk space.

Test Executed:

1) # cd /etc/cron.daily
2) # cat logrotate
3) # cat /etc/logrotate.conf

Expected Output:

1) Verify /usr/sbin/logrotate and /etc/logrotate.conf entries exist, indicating the log rotation utility is used.
2) Ensure that log files are rotated weekly, that empty log files are created after rotating old ones, and 4 weeks of backlogs are kept.

Test Result: Pass.

Risk: If log files are not rotated off the system disk space in a timely fashion, there is risk of consuming the entire disk space, degrading the performance of the server and potentially causing a denial-of-service to legitimate users. Also, if disk space were consumed, new log files would be prevented for being written. This setting is configured properly; the risk associated with setting is **Negligible.**

Recommendation: None.

4.47 System Backup

No matter how much security protection is afforded to protect a system from attackers, accidents, bugs and natural disasters cannot be predicted, and cannot be prevented. The best defense against losing system data is to have a routine

[6] *Swatch http://swatch.sourceforge.net/*

system and data backup schedule. The time and money it would take to rebuild a system from scratch by far outweighs the time and money it takes to maintain system and data backups.

Test Executed:

Discuss with the system administrator the scheme used for backing up the server and its data. Inspect the area designated for storage of the backup media, and the media itself.

Expected Output:
Backups of the server and its data are conducted on a regular basis. Backups are stored in a secure area on read-only media. Backups are sent off-site to a secure facility for storage.

Test Result: Pass.

Risk: If the system was not accessible, or if data was destroyed, without a current backup, there is no means to retrieve that data. Critical or sensitive data may be overwritten or lost, or newly stored data may be corrupted by some residual data on the reallocated media. System and data backups are performed regularly; backups are secured on-site and copies are stored off-site; the risk associated with setting is **Negligible.**

Recommendation: None.

4.48 Disaster Recovery Plan

A Disaster Recovery Plan is a vital piece to the overall security posture of an information system. A Disaster Recovery Plan specifies the procedures for recovering normal business operations in the event of a crisis situation.

Test Executed:

Document review of the system Disaster Recovery Plan.

Expected Output:

A complete and current disaster recovery plan is in place. The plan contains the following attributes:

- The plan deals with critical events (e.g., the loss or damage of critical files or equipment to the destruction of the entire facility) and includes actions to continue processing.
- The plan assigns responsibility to employees involved in disaster recovery efforts.

- The plan provides methods to continue operations, including the use of an alternate processing facility, which has been designated.

Test Result: Not Met. A formal Disaster Recovery Plan is not in place.

Risk: If a disaster or emergency were to occur, and the environment has no planned method for recovery, actions could take place that would cause recovery to be costly, unorganized, and detrimental to the security of the information housed in the environment. Personnel would not be aware of what to do in an emergency, and in the process the system could be irrevocably damaged, and personal safety could be jeopardized. Systems and business operations would not be able to be resumed in a timely fashion, potentially rendering employees unable to complete work. The risk of exploitation of this finding is **High.**

Recommendation: Draft, finalize and implement a formal, written Disaster Recovery Plan that establishes procedures for resuming business operations in the event of a crisis, and establishes roles and responsibilities of employees.

References

Red Hat Linux Security Guide
http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/

The Linux Systems Administrator Guide
http://www.tldp.org/LDP/sag/html/x2869.html

IT Security Cookbook – Securing UNIX #1
http://www.boran.com/security/unix1.html

IT Security Cookbook – Securing UNIX #2
http://www.boran.com/security/unix2.html

Linux Security
http://www.linuxsecurity.com

Garfinkel, Simson and Spafford, Gene. Practical UNIX & Internet Security. O'Reilly & Associates, Inc., April 1996.

Huasin, Kamran and Parker, Timothy. Linux Unleashed, Second Edition. SAMS Publishing, 1996.