Global Information Assurance Certification Paper

Security Audit of GIAC Enterprises' Central Log Server

Rick Robinson
May 20th, 2003
GCUX Practical Assignment v1.9 Option 2

Abstract/Summary

This paper covers the audit of GIAC Enterprises' central log server as performed by Royberson Security Services. During the course of the audit several steps were taken including physically inspecting the site, interviewing employees responsible for administering the log server, and running various security tests on the machine itself. The results both good and bad were then presented in this report along with recommendations on how to resolve the findings.

Table of Contents

Executive Summary

On May 20th, 2003 Royberson Security Services was contracted to perform a security audit on the GIAC Enterprises central logging server. During the audit we discovered several very good security practices already in place within GIAC. Most of the security findings were under the High risk level and all of them can be easily resolved.

The audit consisted of a physical inspection of the site, interviews with Mike McDonough and his security administration team, and multiple server scans and tests to get a picture of the overall state of security on the system. The following is a high level overview of our findings.

- Security patches that could lead to remote access or elevated privileges on the server should be installed immediately
- User accounts and access codes should be removed and changed when employees are terminated or their job functions change and no longer require them to have access to the system
- Log information being sent across the network should be encrypted
- Host based firewall rules should be setup to only allow access to the server from authorized machines
- Configuration changes should be made to SSH to improve security on the connections made to the machine.

Overall we found security awareness within the company to be very good. Strong policies in many areas are already in place and the administration to responsible for the server recognize the importance of protecting the data stored on the machine.

We feel that the recommended changes to the system will produce a security posture consistent with the goals of GIAC Enterprises and will ensure the reliability and integrity of the log information stored on the server.

System Description

Client Overview

GIAC Enterprises is an e-business specializing in the sale of fortune cookie sayings. Their success has continued to grow over the past few years and GIAC now holds 83% of the fortune cookie saying market share. However being a major player in the online fortune cookie business also makes your system a prime target for computer criminals, script kiddies, and even competing businesses. With increasing reports of computer crime across the globe GIAC decided last year it needed to increase it's security posture.

Several improvements were made over the following year to the GIAC network. Firewalls were installed to protect the internal network from the Internet, user awareness policies were created to educate employees within the company on the importance of using strong passwords[1], how to recognize and respond to social engineering attempts, and procedures were put in place to regularly review security vulnerabilities and patch security holes in a timely manner.

The most recent addition to the GIAC security platform is a central logging server[2]. The server has been running for a few months now and GIAC is happy with it's performance so far, but considering how sensitive the information being sent to the server is and the importance that server has in being able to investigate security incidents they wanted to make sure it was locked down air tight. So Royberson Security Services was contracted to come out and audit the security of the central logging system.

System Specifications

Despite their recent success GIAC Enterprises is still a pretty small company in the grand scheme of things. So any decisions they make in terms of purchasing equipment, and implementing security solutions are always made with cost being a major consideration. The central log server was no different. The machine that became the log server was an old Linux web server that was taken out of service when the company consolidated several of their web services onto a single host. GIAC prefers to implement open source software solutions whenever possible. Preferably software released under a GPL or BSD type license.

The following is a description of the server being audited. It details the current hardware platform and operating system the server is running on, along with some of the more important software packages its running and the revision numbers of those packages.

---

[1] National Infrastructure Protection Center. "Password Protection 101". URL: http://www.nipc.gov/publications/nipcpub/password.htm (20 May 2003).
[2] Hines, Eric. "Complete Reference Guide to Creating a Remote Log Server". 22 Aug. 2000. URL: http://linuxsecurity.com/feature_stories/feature_story-64.html (20 May 2003).

Server Details

Hostname	logger.giacfc.com
Operating System	Red Hat Linux 7.3
Processor	Intel Pentium III 800 MHz
System Memory	256 Megabytes
Disk Space	2 WDC 120 Gigabyte IDE

Software Packages

Name	Version
OpenSSH	3.1p1, SSH Protocols 1.5/2.0, OpenSSL 0x00090602f
OpenSSL	0.9.6b
Sendmail	8.11.6
Syslog	1.4.1

Logger acts as a central log repository for all the Unix and Unix-like servers within the GIAC network. Logs are forwarded to Logger from the remote hosts via the Syslog daemon. The extra security on this machine provides the system administrators with a set of known good logs. Since the logs are sent in real time even if an attacker were to compromise a remote system and destroy the logs to cover his tracks a copy of his activities would be sent to the central log server.

Anything that could compromise the integrity of the data being sent to the server would be the biggest concern in this system. If the information is tainted then it renders the log server useless. The GIAC network is behind a firewall so only machines on the corporate network can communicate with the server. That greatly reduces the risk of attacks launched from the Internet. Within the network the biggest risks come from employees launching insider attacks, and from network overload that may cause log messages to get lost in transit.

Sendmail[3], OpenSSH[4], and OpenSSL[5] have all had their fair share of root exploits, especially recently. Special focus will have to be paid to these services and to the Syslog service since they are the main interaction points with the server. Any security vulnerabilities that are found in any of these programs should be considered High Risk and dealt with immediately.

Audit Methodology

The audit was conducted in multiple stages. In the first stage we performed a physical inspection of the site where the server was located. Michael McDonough

[3] Sendmail Consortium. *Sendmail* URL: http://www.sendmail.org (20 May 2003).
[4] OpenBSD Project. *OpenSSH*. URL: http://www.openssh.org (20 May 2003).
[5] OpenSSL. *OpenSSL*. URL: http://www.openssl.org (20 May 2003).

represented GIAC Communications and oversaw the inspection. Mr. McDonough is the Senior Systems Administrator for GIAC and was responsible for the implementation of the central log server. We looked for what physical security measures were currently in place to protect the server, and what areas could be improved upon.

The second stage of the audit consisted of interviews with Mr. McDonough and the rest of his team who are responsible for the administration of the server. During our discussions with the security administration group we were given a good understanding of the day to day administration tasks required on the server, what type of access is granted to users, what processes are in place to patch security holes, and which roles each of the members played in working with the server.

The final stage of the audit was comprised of several tests to determine the current state of security on Logger. We were given temporary root access and permission to install several security tools to gather information on the system. One of the system administrators was assigned to assist us with the audit and monitor our activities when logged into the server as root.

Two different utilities were used to scan the system. The port-scanning package Nmap[6] was used to perform a non-intrusive scan of the listening services on each of the servers. Other vulnerability scanners incorporate similar functionality, but Nmap provides an excellent way to quickly generate a high level snapshot of the current state of the network. Without having to authenticate into the server we can already ascertain security concerns and at the same time give the client an idea of what type of information an attacker outside of the system can gather.

The vulnerability scan was performed using Nessus[7]. Nessus is far more useful than your typical vulnerability scanner. Most vulnerability scanners base their reports off of banners and version numbers. Nessus attempts to go beyond that by trying to exploit actual code against the services running on your machine. This gives the administrator a far better understanding of what type of risk they are at and what areas need to be addressed.

In addition to the scans we reviewed many of the configuration settings on the machine, permissions on critical files and directories, authentication controls on the system, etc. A more detailed explanation of the tools used and the type of information gathered will be given in the Detailed Analysis section of this report. Any findings will be defined into three different categories:

High risk items should be resolved immediately. They are security risks that could allow an attacker to remotely gain access (root or otherwise) to your system, allow an attacker to gain root access locally on your system, or would

[6] Fyodor. *Nmap Stealth Port Scanner*. URL: http://www.insecure.org/nmap (20 May 2003).
[7] Nessus Project. *Nessus*. URL: http://www.nessus.org (20 May 2003).

allow an attacker to gain control of any system critical processes running on your system. Exceptions should be sought to company change control procedures to get high risk security items resolved at once if they are outside of the normal installation window.

Medium risk items are vulnerabilities that could lead an attacker to gain non-privileged access to your system locally, allow an attacker to alter any non critical data outside of the company change management policies, or would give an attacker additional access to the system or elevated access that is not root, but is above what they were given. These items do present a serious risk to the system, but could be resolved in a less urgent time frame than high risk items. They should be resolved as quickly as possible, but could wait to be resolved to coincide with normal change management windows.

Low risk items are generally vulnerabilities that do not lead to security compromises themselves, but could provide an attacker with information that would allow them to exploit other weaknesses on the system. They would also include attacks or weaknesses that are not of significant risk to the company because of other mitigating controls or because the difficulty or likelihood of successfully exploiting the vulnerability.

All security risks should be addressed whenever possible. However if it is not practical to resolve low risk items, in general they can be left alone without presenting a significant risk to the system.

Detailed Analysis

Any findings will be listed in the body of the section they were discovered under. Each finding will identify the issue, define the risk level, provide a solution to resolve the finding, and provide any CVE[8] or CERT[9] document names that are associated with the vulnerability. Vulnerability descriptions for Red Hat packages were taken all or in part from the Red Hat Errata page detailing Red Hat 7.3 updated security packages[10].

Example:

Finding 1:	The telnet daemon has a buffer over flow that could lead to a remote user gaining root access on the machine.
Risk Level:	High
Solution:	Update the telnetd package to the latest version. Or disable and remove telnet from the system and use SSH.
CVE:	CVE-1924-0001

Physical Security

Upon arriving at GIAC Enterprises we were greeted in the lobby by the receptionist, asked to sign in, and present a piece of photo identification. Once our appointments were verified we were met by Mike McDonough who escorted us back into the data center. Access to the data center requires a sensor badge.

The central log server was located within the server room of the data center. Access to the server room is given with a pass code on a keypad on the door. The construction of the server room looked very solid. The door leading into the room was heavy and solid wood, and we were able to verify that despite the existence of a drop ceiling for cable runs the exterior walls of the server room did extend past the ceiling.

Finding 1:	The server room access code is not changed on a regular basis.
Risk Level:	Medium
Solution:	The pass code to access the server room should changed on a regular basis. It should be changed any time an employee who knew the pass code was terminated. Only employees who need access to the servers should know the pass code.
CVE:	None

[8] The MITRE Corporation. "Common Vulnerabilities and Exposures". 27 Mar. 2002. http://cve.mitre.org/cve/ (20 May 2003).
[9] Carnegie Mellon Software Engineering Institute. "CERT Coordination Center". http://www.cert.org (20 May 2003).
[10] Red Hat, Inc. "Red Hat Linux 7.3 Security Advisories". https://rhn.redhat.com/errata/rh73-errata-security.html. (20 May 2003).

Logger was located inside of a key locked rack and in addition to the rack being locked access to the floppy drive, tape drive, and power buttons were locked on the server itself. The security administration team members are the only ones who have keys to unlock the server door. The combination of badge access, keypad access, and key access to the server makes for a very solid physical security setup.

Operating System Vulnerabilities

Like most Unix and Unix-like systems a Red Hat Linux installation tends to install more packages than what the server requires. Some of these unused packages could develop security holes and put the system at risk. The security administration team should review all of the packages installed on the system and remove any unused packages. A list of all RPM packages can be found by running the following command:

/bin/rpm –q –a

And unused packages can be removed by running the following command:

/bin/rpm –e <package name>

Once all unnecessary packages have been removed from the system the Red Hat Errata page[11] should be reviewed to determine if any installed packages have security vulnerabilities that need to be upgraded. For this audit we chose the Security Alerts link for Red Hat 7.3. Some of the vulnerability fixes may be listed in the Risks from third-party software section later in this report. Only High and Medium risk vulnerabilities will be reported in this audit. Please review the Errata page for the full list of vulnerabilities.

Finding 2:	The util-linux package contains a locally exploitable vulnerability that could lead to root privileges on the system.
Risk Level:	Medium
Solution:	Update to the latest version of the following packages: losetup, mount, and util-linux. This finding was set at a medium level even though it could lead to root privileges because of the difficulty in executing this attack.
CVE:	CAN-2002-0638

[11] Red Hat, Inc. “Errata: Security Alerts, Bugfixes, and Enhancements”. http://www.redhat.com/apps/support/errata (20 May 2003).

Finding 3:	Multiple security holes in the Linux kernel could lead to denial of service attacks or root privileges for local users.
Risk Level:	High
Solution:	Update the system kernel to 2.4.20 or later.
CVE:	CAN-2003-0244, CAN-2003-0246

Finding 4:	An integer overflow is present in the xdrmem_getbytes() function of glibc 2.3.1 and earlier. Depending upon the application, this vulnerability could cause buffer overflows and may be exploitable leading to arbitrary code execution.
Risk Level:	High
Solution:	Update the glibc packages to version 2.2.5-43
CVE:	CAN-2003-0028

Security Patch Installation and Management

The security administration team just recently put procedures in place to regularly review security advisories and handle installing security patches. Change management policies for GIAC Enterprises require all code changes to be made within scheduled releases every month. Any package updates done outside of the monthly release would have to get approval from senior management. Senior management is willing to sign off on exceptions for High risk security vulnerabilities.

The new review procedures include members of the security administration team reviewing the Red Hat Errata web site on a daily basis, reviewing discussions on security mailing lists like Bugtraq[12] and the Red Hat Watch List[13]. You can subscribe to Bugtraq and several other security related mailing lists at http://www.securityfocus.com/subscribe and you can subscribe to the Red Hat Watch List at http://www.redhat.com/mailman/listinfo/redhat-watch-list. Any time a security vulnerability is discovered it is assessed for risk and installed into the system accordingly. High risk vulnerabilities that fall outside of the monthly release window will go through the exception process. Medium risk vulnerabilities will be scheduled to be installed with the next monthly release, and Low risk vulnerabilities will be installed as deemed necessary.

GIAC has a fully functional test lab where all patches will be tested before they are put into the production environment. Once the software update has gone through sufficient regression testing and no issues have surfaced it is approved to be installed into production. There is approximately a 1 week turnaround on regression testing for most applications.

[12] Security Focus. "Bugtraq Full Disclosure Security Mailing List". http://www.securityfocus.com/archive/1 (20 May 2003).
[13] Red Hat, Inc. "Red Hat Watch List – Security and Bugfix announcements". http://www.redhat.com/archives/redhat-watch-list/ (20 May 2003).

Configuration Vulnerabilities

Passwords

Password controls on the central logging server are pretty strong. The default configuration has all encrypted password strings housed in the /etc/shadow file and passwords are encrypted using MD5 which makes it more difficult for an attacker using a program such as John the Ripper[14] to crack the passwords. In addition the following password options are configured in the /etc/login.defs file:

```
# Max number of days a password can be used
PASS_MAX_DAYS        90
# Min number of days between password changes
PASS_MIN_DAYS        5
# Min acceptable password length
PASS_MIN_LEN         8
# Number of days warning given before a password expires
PASS_WARN_AGE        7
```

The pam module pam_passwdqc[15] is used to perform password strength checking when the passwd program is called. passwdqc will also check the minimum length of a password and will check to ensure it has the user defined number of alphanumeric and special characters.

There are pretty strict controls in place for the root password. The GIAC Enterprises network shares password file between hosts. However with the increased security the central log server maintains it's own password file and it's own root password. The only employees who have access to the root password are the members of the security administration team. The root password is changed by the security administration team every month.

OpenSSH

The /etc/ssh/sshd_config file contains all the ssh server configuration settings. In a secure environment such as a central log server there are certain settings that should be changed to increase security with SSH transmissions. Protocol 1 support should be disabled in the file be specifying only Protocol 2 support. PermitRootLogin should be set to no. PasswordAuthentication should be set to no. Public key authentication is a more secure way of handling SSH transmissions and should be used in place of passwords in this type of environment. The banner directive should be set to a file that informs users making connections to the system that unauthorized access is forbidden and that all activity is subject to monitoring.

[14] Openwall Project. *John the Ripper*. http://www.openwall.com/john/ (20 May 2003).
[15] Openwall Project. *pam_passwdqc*. http://www.openwall.com/passwdqc/ (20 May 2003).

Finding 5:	The outlined changes to the sshd_config file have not been made. Currently root can ssh into the system and users can authenticate using passwords with the weaker protocol 1.
Risk Level:	Medium
Solution:	Make the outlined changes to the sshd_config file. PermitRootLogin should be set to no and PasswordAuthentication should be set to no. Make sure to create your SSH keys before you disable password authentication.
CVE:	None

Insecure Services

The classic insecure services such as rsh, rlogin, and telnet have all been disabled in the xinetd configurations. These services send information across the network in clear text and all of these services can be handled with an encrypted transmission using SSH.

In addition to the unneeded services being disabled in the xinetd configs all unneeded boot scripts have been removed from the /etc/rc.d/rc2.d and rc3.d directories. This will stop unneeded services from starting up at boot time.

File Permissions

Even though only the security administration team has authorized shell access to the central log server all log files should have file permissions of 640 and should be owned by root:security. Then any users who need read access to the log files can be added into the security group. The following command will verify whether or not the files in the logs directory have the proper permissions.

find /var/log –type f –perm –640 –user root –group security –ls

In addition to ensuring proper permissions on the log files, the system administrators should verify there are no unnecessary or improperly set SUID/SGID files on the system. The following command will identify all SUID/SGID files. Any files that do not need the SUID or SGID bit set should have those bits removed with the chown command:

find / -type f \(-perm –2000 –o –perm –4000 \) –ls

You can remove the SUID bit with the following command:
chown u-s <filename>

And the SGID bit can be removed with this command:
chown g-s <filename>

Risks From Installed 3rd Party Services

Our two system scans helped identify 3rd party services that were installed and running on our central log host. We first ran an Nmap scan to determine what services could be seen on the corporate network.

nmap –sT –sU –sR –PT –O logger.giacfc.com

Starting nmap 3.27 (www.insecure.org/nmap/) at 2003-05-20 02:03 EDT
Interesting ports on logger.giacfc.com:
(The 3090 ports scanned but not shown below are in state: closed)
Port State Service (RPC)
22/tcp open ssh
514/udp open syslog
1241/tcp open msg
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.210 days (since Sun May 19 21:01:32 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 11.720 seconds

The Nmap results tell us that we only have 3 services listening on the network. All three services were expected when the scan was run. The SSH and Syslog services are self explanatory. The service running on port 1241 is our Nessus server, which we will take a look at next. Notice that in addition to the services running Nmap was bale to correctly identify that the central log server was running a version of Linux. This type of information can be very useful to attackers who are targeting specific operating systems.

Next we ran a Nessus scan on the server. This scan will give us more detail on areas where we may have security vulnerabilities. We will only list the high risk items from the vulnerability scan. Please reference Appendix A for the full report.

Nessus Report

Hole #1

Service: ssh (22/tcp)
Severity: High

You are running a version of OpenSSH older than OpenSSH 3.2.1

A buffer overflow exists in the daemon if AFS is enabled on your system, or if the options KerberosTgtPassing or AFSTokenPassing are enabled. Even in this scenario, the vulnerability may be avoided by enabling UsePrivilegeSeparation.

Versions prior to 2.9.9 are vulnerable to a remote root exploit. Versions prior to 3.2.1 are vulnerable to a local root exploit.

Solution :
Upgrade to the latest version of OpenSSH

Risk factor : High
CVE : CAN-2002-0575
BID : 4560

This particular hole does not seem to affect the GIAC server. Neither KerberosTgtPassing nor AFSTokenPassing are enabled in the /etc/ssh/sshd_config file. If this were the only SSH alert on the report some consideration should still be given to upgrading to the latest version of SSH. However it would not need to be considered a High risk situation.

Hole #2

Service: ssh (22/tcp)
Severity: High

You are running a version of OpenSSH which is older than 3.4

There is a flaw in this version that can be exploited remotely to give an attacker a shell on this host.

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.

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

Returns :
openssh-server-3.1p1-6

Solution : Upgrade to OpenSSH 3.4 or contact your vendor for a patch
Risk factor : High
CVE : CAN-2002-0639, CAN-2002-0640
BID : 5093

Here are the results of the rpm command the alert suggested we run.

rpm –q openssh-server
openssh-server-3.1p1-3

Finding 6:	There is a flaw in OpenSSH that could be exploited remotely and allow an unauthorized user to gain shell access to the system
Risk Level:	High
Solution:	Upgrade to the latest version of SSH. By doing so we will also be resolving the security issues from the previous Nessus alert.
CVE:	CAN-2002-0639, CAN-2002-0640

You may have noticed that we originally listed Sendmail in the list of 3rd party software packages, but it didn't show up on either of the system scans. That is because Sendmail is configured to only respond to requests from the local host. GIAC only uses Sendmail on that machine to send out pages from e-mail so it will not process connection requests from outside of the box.

It appears there are no other known security vulnerabilities with the installed 3rd party software. It is very important to keep on top of security vulnerabilities with these 3rd party packages. Especially with the services that are listening on the network. Any time a High risk vulnerability develops in SSH, Syslog, or Nessus it should be resolved at once.

Administrative Practices

The security administration team has developed a suite of custom tools to help keep up with the day to day security administration on the central server. Each of these scripts were written in Perl[16]. These scripts are run nightly from cron and perform various security checks for the team. Some of the checks include scanning for the existence of new SUID/SGID files, reviewing the authorization logs for any excessive invalid login attempts, and watching for unauthorized users being added to the password file. Each of these scripts produces a daily report, which is viewed every morning, and any exceptions found on the reports are followed up on immediately.

In addition to the custom daily reports the security administration team runs Tripwire[17] nightly on the system to watch for any unauthorized changes to the system. Tripwire takes a snapshot of the system and then reports when any files have been altered from that snap shot picture. While it is strictly a reactive system it is very useful for quickly identifying changes to the system. Tripwire is also used in the change management procedures when custom code is being installed from the vendor. The vendor provides a list of all files that will change

[16] Larry Wall. *Perl*. http://www.perl.com (20 May 2003).
[17] Tripwire. *Tripwire*. http://sourceforge.net/projects/tripwire (20 May 2003).

when the code is implemented and that list is compared with the Tripwire report to ensure only the code that was expected was installed.

The members of the team all share the root password and use it whenever privileged commands need to be run on the system. This makes creating an audit trail of which users ran which privileged commands very difficult. In addition using the root account can be very dangerous. It's easy to mistype some commands and cause serious problems on the system.

Finding 7:	All members of the Security Administration team know the root password and use it when their job responsibilities require them to have privileged access.
Risk Level:	Medium
Solution:	GIAC Enterprises should consider converting the users to Sudo[18]. Sudo allows the system administrator to define certain commands a user can run with elevated privileges. In addition to being able to select which commands a user runs all Sudo commands are logged which makes creating an audit trail much easier.
CVE:	None

Identification and Protection of Sensitive Data on the Host

The log files from the various Unix servers are the main source of sensitive information on the server. Log files can contain information that would aid an attacker in gaining access to the system. All log files are currently stored under the /var/log directory and are all owned by root:security with 640 permissions. This keeps anyone other than root and users in the security group from accessing these files.

Some consideration should be given to storing the log files encrypted on the disk. Tools like GnuPG[19] would help ensure the data is protected even if the system were to be compromised. However if the data is stored encrypted it will greatly slow down the process of accessing that data.

Finding 8:	Encrypt log files while being stored on the central log server
Risk Level:	Low
Solution:	Use an encryption tool such as GnuPG to store the log files encrypted on the server. While encrypting the files would add a strong layer of protection, the fact that access is already limited on the system keep this from being a Medium risk.
CVE:	None

[18] Todd Miller. *Sudo*. URL: http://www.courtesan.com/sudo/sudo.html (20 May 2003).
[19] GNU Privacy Guard. *GnuPG*. URL: http://www.gnupg.org/ (20 May 2003).

The only other truly sensitive data on the server would be the user authentication information. With the password being stored in the /etc/shadow file which is only readable by root, and with those passwords being stored as MD5 encrypted strings adequate controls exist to protect user access information.

Protection of Sensitive Data in Transit Over the Network or Internet

There are no data transmissions going to or from the central log server that pass over the Internet. The corporate firewalls keep all data stream contained within the corporate network.

There are only two ways to communicate remotely with the central log server. The server listens for requests from SSH on TCP port 22 and for Syslog messages on UDP port 514. Currently it also listens on TCP port 1241 for Nessus communications, but that could be disabled.

All remote administration is handled through SSH. SSH encrypts the data streams and helps ensure that attackers on the network can't sniff[20] sensitive data such as authentication information. SSH also has stronger authentication controls with the use of public key authentication[21]. In order to authenticate on the system the attacker would need to have a copy of the users private key, and the pass phrase to use that private key. Administration activities for this application can be considered secure.

The other interface into the log server is through Syslog. These transmissions are not secure. All log messages are sent across the network in clear text and could be intercepted and read by attacker on the network. In addition an attacker could potentially perform a man in the middle attack where they intercepted the data, altered it to hide their tracks, and then sent it on to the log server. Since Syslog also uses the UDP protocol there is the potential that data could be lost in transit. Unlike TCP packets, if a UDP packet is dropped in transit it is not resubmitted to the server.

[20] Graham, Robert. "Sniffing (network wiretap, sniffer) FAQ". Version 0.3.3. 14 Sep. 2000. URL: http://www.robertgraham.com/pubs/sniffing-faq.html (20 May 2003).
[21] Whittle, Robin. "Public Key Authentication Framework: Tutorial". 2 Jun. 1996 URL: http://members.ozemail.com.au/~firstpr/crypto/pkaftute.htm (20 May 2003).

Finding 9:	Log files are sent across the network to the central log server in clear text.
Risk Level:	Medium
Solution:	There are now alternatives to the standard Syslog daemon. One of these is nsyslog[22], which supports TCP and SSL encryption. The recently released SDSC Syslog[23] a drop in replacement that not only supports TCP and encryption, but can also handle standard UDP/514 Syslog transmissions. This is a great feature because it doesn't force network administrators to convert the entire site to a new Syslog format all at once. This is considered a medium security risk because of the importance of ensuring the integrity of the log files sent to the server.
CVE:	None

The server also has a small real time alert system that generates outbound e-mail traffic when the server load hits a certain threshold or disk space is running short. The information contained within these e-mails is generally not very sensitive, but to be on the safe side the security administration team has used GnuPG to encrypt these e-mails before they are sent out.

Access Controls

Access to the server is only given to members of the security administration team. Each member is given shell access to the machine and also knows the root password to perform privileged tasks. Even though the sshd_config file does not disallow remote root logins, the members of the group have all been added to the wheel group on the system and generally log in as themselves and then su to root.

Finding 10:	Two accounts that belonged to employees who had been terminated were still active on the system.
Risk Level:	Medium
Solution:	During a review of the accounts in the /etc/passwd file it was noted that two former members of the security administration team still had access to the server. Procedures should be put in place to remove user accounts when an employee leaves the security administration team.
CVE:	None

While the corporate firewall does protect the server from connections originating from the Internet, the central log server does not filter out any connection attempts originating from anywhere within the corporate network. Even though

[22] Darren Reed. *nsyslog*. http://coombs.anu.edu.au/~avalon/nsyslog.html (20 May 2003).
[23] San Diego Supercomputer Center. *High Performance Syslog*. http://security.sdsc.edu/software/sdsc-syslog/ (20 May 2003).

they do not have accounts on the system, employees who have access to the local network can use tools such as Nmap to scan the central log server and potentially find weaknesses to exploit.

Finding 11:	The central log server does not filter out SSH and other connection requests originating from other parts of the corporate network.
Risk Level:	Low
Solution:	Netfilter[24] should be implemented on the server to block connection attempts from unnecessary machines on the network. Iptable rules should be put in place to only allow SSH connections from the workstations of the security administration team and only allow Syslog messages to be sent from the Unix servers. All other connection attempts should be denied. This will help mitigate the risks when a security vulnerability is discovered in one of these services.
CVE:	None

Disaster Recovery Procedures

The security administration team has a solid disaster recovery plan already in place. Full tape backups are made of the server once a month and incremental backups are made every day. All tapes are rotated off site to a secure 3rd party storage facility. The daily tapes are rotated back into use every two weeks and the full monthly backups are kept for a minimum of two years.

In addition to creating the backups, the administration team has recovery drills once every quarter. Data loss is simulated in the test environments and the service is restored from the backup tapes.

Other Issues

A final concern with this system is there are no processes in place to perform an entire OS upgrade at some point in the future. Assuming that security patches are kept up with on a regular basis the risks associated with this are low, but Red Hat will at some point stop supporting the 7.3 platform. At that point they will no longer produce security packages for 7.3.

Finding 12:	No policy is in place to perform an OS upgrade after a specified time period.
Risk Level:	Low

[24] Netfilter. *Netfilter*. http://www.netfilter.org/ (20 May 2003).

Solution:	Create a policy to upgrade the entire OS after a specified time period. Possibly every 24 months.
CVE:	None

Critical Issues and Recommendations

Top Ten System Vulnerabilities

1. **A remote vulnerability exists in the version of OpenSSH currently installed on the system**

   The Nessus security scan performed on the local system discovered a remote exploit that could allow an attacker to gain access to the server. If the attacker were able to get shell access they could then look for other local exploits gain elevated privileges on the machine or at the very least perform any tasks and alter any data the owner of the process has access to.

   Any services listening to the outside world that develop security vulnerabilities should be considered high risk. Red Hat has released updated RPM packages for OpenSSH.

2. **Multiple security holes in the Linux kernel could lead to a local user gaining root privileges or causing a denial of service attack**

   The Linux kernel is the heart of the system. All Linux distributions are built around the same kernel. There were several security holes discovered in an audit of the 2.4 kernel code. These holes have been resolved in versions of the kernel numbering 2.4.20 and later.

   An updated kernel can be retrieved in RPM package form from the Red Hat Errata site or the source code can be downloaded and compiled from ftp.kernel.org. Many administrators choose to compile their kernel from source because it allows them to fine tune the kernel for their hardware.

3. **A buffer overflow exists in one of the functions within glibc that could lead to arbitrary code being executed**

   Just like the previous two outlined security vulnerabilities, Red Hat has updated packages on their Errata site to resolve this vulnerability. You can update your package with the following command:

   # rpm –Uvh <Package name>

4. **The access code to the server room is not changed on a regular basis**

The keypad access code needed to enter the server room should be changed regularly. The thought process behind this is if an attacker were to discover the code to access the room it would hopefully be changed before he had a chance to use it or at least lock him out when he tries to access it in the future.

It is also very important that the access code is changed whenever an employee who had access to the server room is terminated or their job duties no longer require them to access the server room. Physical access to the server at the very least could result in a denial of service attack if the attacker cuts power to the machine, but also could result in an attacker gaining access to the machine by causing it to reboot and logging in through single user mode.

5. **The util-linux package contains a vulnerability that could give a local attacker elevated privileges**

   The util-linux package contains several useful utilities for system administration. A flaw was discovered that could allow a local attacker to gain root privileges on the machine.

   This finding was downgraded to a medium risk because of the sheer difficulty in successfully exploiting this hole. However it should still be taken seriously and fixed as quickly as possible. Updated RPMs can be found on the Red Hat Errata site.

6. **Several changes should be made to the /etc/ssh/sshd_config file**

   Remote root logins should never be allowed on the system. If a user needs to be able to access the system as root they should first log into the system with their own account and then su to root from there. Disallowing root logins ensures that even if an attacker were to discover the root password they would still need access into the system before they could use it.

   Password authentication should be disabled on the server. Using SSH keys creates a two factor authentication system, which is more secure than standard password authentication. With SSH keys you must have a copy of the users private key and the pass phrase to load that key in order to authenticate. Passwordless SSH keys should never be used with interactive user accounts.

   A banner file should be created alerting any potential attackers that unauthorized access to the system is forbidden and that all activities on the system are subject to monitoring. This gives the attacker a chance to

disconnecting before they have accessed the system and gives the company some legal footing should they discover and prosecute the attacker.

All of these changes are made by altering the corresponding entries in the sshd_config file and then restarting the sshd process. Note: You can restart the sshd process by sending it a –HUP signal. This will not disconnect existing ssh connections so changes can be made even when users are logged into the system.

7. Members of the security administration team share the root password and use it to perform required privileged tasks

The root password should only be given to users on a need to know basis. Logging into root to perform privileged functions does not allow an audit trail to be created of what privileged commands have been run on the system. Users who must run commands as root or as other users should be converted to using Sudo. Sudo allows a set of commands to be defined that can be a run by a given user with elevated privileges. In addition these commands are all logged and can be reviewed later. Even users who need access to the root password should be converted to using Sudo for most of their tasks.

By using Sudo a procedure can also be created to review incidents where su to root was used. Since the legitimate users on the system can use Sudo to complete their tasks an su record in the logs could indicate a system compromise.

8. Log files are sent across the network to the server in clear text

The purpose of a central log server is to ensure the integrity of the data being reviewed. By sending information across the network in clear text that data could be compromised. It's possible an attacker could intercept that data and from it gain information that would allow them to compromise either the log server itself or another system on the network. It's also possible that an attacker who has compromised the system could perform a man in the middle attack and alter the data on the way to the server to hide their tracks.

There are multiple alternatives to the standard Syslog that can be installed. Our recommendation is to use the SDSC's High Performance Syslog. Not only does it support sending Syslog messages encrypted over TCP, but it also integrates seamlessly with standard Syslog daemons. This means machines could slowly be converted to the new Syslog format as is practical for the site. The SDSC Syslog also has been written to

more effectively handle the load for busy sites than standard Syslog. The source code can be obtained and then compiled from the SDSC web site.

9. **Accounts were found on the system belonging to employees who had been terminated from the company**

   All system accounts belonging to an employee should be immediately removed if that employee is terminated or if their job duties no longer require them to access the system. This helps protect against disgruntled employees from attacking the system.

10. **The log server does not filter out connection attempts from machines out side of the security administration team and the Unix servers**

    Netfilter can be used to create a host based firewall to disallow connection attempts from unauthorized machines. Firewall rules should be setup to only allow SSH connections to be established from the workstations belonging to the members of the security administration team and to allow Syslog connections to be made from the Unix servers.

    This keeps outside users from being able to use tools like Nmap to scan the machine for potential security risks and also helps mitigate the risk of exploits against listening services on the machine. If the server drops all SSH packets from an attacker then the attacker can not take advantage of any vulnerabilities in SSH. However this should not be considered a reason to not upgrade a vulnerable SSH installation. It's possible security vulnerabilities in Netfilter could allow an attacker to bypass the firewall and still attack your services. So keep those services patched.

Final Thoughts

Overall we were very impressed with the security posture of the central log server. There are good controls in place for accessing the machine, reviewing daily logs, and backing up the data. We also suspect that the recently implemented security vulnerability review process will resolve some of the finding we had in the future.

While it is outside of the scope of this audit the security on the machines connecting to the server should also be reviewed. Strict access controls should be placed on those servers as they can act as a bridge into the central log sever. Also the remote Syslog daemons should be configured to use an alternate configuration file and a decoy configuration file should be left in the /etc directory.

Appendix A – Complete Nessus Scan of logger.giacfc.com

NESSUS SECURITY SCAN REPORT

Created 20.05.2003 Sorted by host names

Session Name : Session1
Start Time : 20.05.2003 02:16:30
Finish Time : 20.05.2003 23:19:30
Elapsed Time : 0 day(s) 65534:65480:65477

Total security holes found : 9
high severity : 2
low severity : 6
informational : 1

Scanned hosts:

Name	High	Low	Info
192.168.1.2	2	6	1

Host: 192.168.1.2

Open ports:

ssh (22/tcp)

Service: ssh (22/tcp)
Severity: High

You are running a version of OpenSSH which is older than 3.4

There is a flaw in this version that can be exploited remotely to give an attacker a shell on this host.

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.

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

Returns :
openssh-server-3.1p1-6

Solution : Upgrade to OpenSSH 3.4 or contact your vendor for a patch
Risk factor : High
CVE : CAN-2002-0639, CAN-2002-0640
BID : 5093

Service: ssh (22/tcp)
Severity: High

You are running a version of OpenSSH older than OpenSSH 3.2.1

A buffer overflow exists in the daemon if AFS is enabled on your system, or if the options KerberosTgtPassing or AFSTokenPassing are enabled. Even in this scenario, the vulnerability may be avoided by enabling UsePrivilegeSeparation.

Versions prior to 2.9.9 are vulnerable to a remote root exploit. Versions prior to 3.2.1 are vulnerable to a local root exploit.

Solution :
Upgrade to the latest version of OpenSSH

Risk factor : High
CVE : CAN-2002-0575
BID : 4560

Service: general/tcp
Severity: Low

Remote OS guess : Linux Kernel 2.4.0 - 2.5.20

CVE : CAN-1999-0454

Service: ssh (22/tcp)

Severity: Low

Remote SSH version : SSH-1.99-OpenSSH_3.1p1

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 3.6.1p1 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 an
existent 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

Service: ssh (22/tcp)
Severity: Low

An ssh server is running on this port

Appendix B – References

Articles and Books Referenced in the Report and Used in Research

1. National Infrastructure Protection Center. “Password Protection 101”. URL: http://www.nipc.gov/publications/nipcpub/password.htm (20 May 2003).
2. Hines, Eric. “Complete Reference Guide to Creating a Remote Log Server”. 22 Aug. 2000. URL: http://linuxsecurity.com/feature_stories/feature_story-64.html (20 May 2003).
3. The MITRE Corporation. “Common Vulnerabilities and Exposures”. 27 Mar. 2002. URL: http://cve.mitre.org/cve/ (20 May 2003).
4. Carnegie Mellon Software Engineering Institute. “CERT Coordination Center”. URL: http://www.cert.org (20 May 2003).
5. Red Hat, Inc. “Red Hat Linux 7.3 Security Advisories”. URL: https://rhn.redhat.com/errata/rh73-errata-security.html. (20 May 2003).
6. Red Hat, Inc. “Errata: Security Alerts, Bugfixes, and Enhancements”. URL: http://www.redhat.com/apps/support/errata (20 May 2003).
7. Security Focus. “Bugtraq Full Disclosure Security Mailing List”. URL: http://www.securityfocus.com/archive/1 (20 May 2003).
8. Red Hat, Inc. “Red Hat Watch List – Security and Bugfix announcements”. URL: http://www.redhat.com/archives/redhat-watch-list/ (20 May 2003).
9. Graham, Robert. “Sniffing (network wiretap, sniffer) FAQ”. Version 0.3.3. 14 Sep. 2000. URL: http://www.robertgraham.com/pubs/sniffing-faq.html (20 May 2003).
10. Whittle, Robin. “Public Key Authentication Framework: Tutorial”. 2 Jun. 1996 URL: http://members.ozemail.com.au/~firstpr/crypto/pkaftute.htm (20 May 2003).
11. Barrett, Daniel J. & Silverman, Richard. SSH, The Secure Shell: The Definitive Guide. O’Reilly & Associates, Inc., January 2001.

Security Tools and Applications

1. Sendmail Consortium. *Sendmail* URL: http://www.sendmail.org (20 May 2003).
2. OpenBSD Project. *OpenSSH*. URL: http://www.openssh.org (20 May 2003).
3. OpenSSL. *OpenSSL*. URL: http://www.openssl.org (20 May 2003).
4. Fyodor. *Nmap Stealth Port Scanner*. URL: http://www.insecure.org/nmap (20 May 2003).
5. Nessus Project. *Nessus*. URL: http://www.nessus.org (20 May 2003).
6. Openwall Project. *John the Ripper*. http://www.openwall.com/john/ (20 May 2003).
7. Openwall Project. *pam_passwdqc*. http://www.openwall.com/passwdqc/ (20 May 2003).

8. Larry Wall. *Perl*. http://www.perl.com (20 May 2003).
9. Tripwire. *Tripwire*. http://sourceforge.net/projects/tripwire (20 May 2003).
10. Todd Miller. *Sudo*. URL: http://www.courtesan.com/sudo/sudo.html (20 May 2003).
11. GNU Privacy Guard. *GnuPG*. URL: http://www.gnupg.org/ (20 May 2003).
12. Darren Reed. *nsyslog*. http://coombs.anu.edu.au/~avalon/nsyslog.html (20 May 2003).
13. San Diego Supercomputer Center. *High Performance Syslog*. http://security.sdsc.edu/software/sdsc-syslog/ (20 May 2003).
14. Netfilter. *Netfilter*. http://www.netfilter.org/ (20 May 2003).