Global Information Assurance Certification Paper

GIAC ENTERPRISES SECURITY AUDIT

Performed by Shane Machon
On Date: 15th – 17th of May 2002

TABLE OF CONTENTS:

Executive Summary:

GIAC Enterprises Ltd recently undertook a full system audit of the key server running the organisation's website.

The purpose of this audit is to analyse, evaluate and to recommend improvements that can be made to the overall security of the server, *diego*, located on the premises of GIAC Enterprises. A graphical representation of the network can be found in figure 1.1 in the following section that outlines the server's role in the organisation.

This audit was performed using several security vulnerability and analysis tools. These tools tested both the external security of *diego* as well as the internal security of the data and OS configuration. The results of the analysis show some major issues with the services running on *diego*, mainly in the DNS implementation (BIND) and the inetd service. Other issues consist of common configuration faults in the Apache webserver and OS vulnerabilities that could result in a system compromise. Most of these issues can be resolved with patches that can be applied by the UNIX Administrator.

The backup system that has been implemented is sufficient in the event of a data or operating system failure, however there is no disaster recovery scenario in place for faulty hardware or other fatal environmental event. Due to the critical nature of the server, it is recommended that a server with similar or if possible the same hardware be purchased in the event of a serious server hardware failure. This will prevent lengthy downtime whilst hardware is shipped from manufacturers/suppliers. As there is no development server on the network, the identical backup server can be implemented as a test bed for the data that is to be placed on *diego*, and also to evaluate system patches of the applications and OS on the primary server. This ensures all changes made to *diego* have been tested on the test bed server before implementation.

The IT staff that maintain the network require additional tools to actively monitor servers and mission critical applications that are running on each server, it is recommended by the audit that a solution be implemented urgently to assist IT staff to monitor these servers and ensure maximum connectivity and uptime.

The existing policies for the administration of the server can also be improved. The implementation of several tools should be reviewed for system administrators to pro-actively view system logs and application logs to identify potential problems that can be found and resolved before it can cause system downtime for the server. It is also recommended that IT staff enlists in services such as CERT or SANS security-related organisations, as well as OS vendors. This is so that they can be made aware of security vulnerabilities that exist with the OS and applications that are running on *diego* and other servers on the network.

Since the servers have only been implemented in the past 12 months, it is recommended that this audit be retaken again in the next 6 months to ensure that all priority issues have been resolved and any further issues are brought to GIAC’s attention.

Section 1 - System and Audit Methodology

1.1 Network Infrastructure

The network at GIAC enterprises is designed around the companies' website, running on *diego*. Other servers consist of a DNS server, that houses the companies primary domain name, as well as internal server name allocation secondary services and handling of the organisation's email. A postgres database server is also on the network to house the backend database for the webserver on *diego*.

These 3 servers are located in a de-militarised zone (DMZ). This area is designed so that developers, network administrators and the website user can interact with the website without compromising the security of the organisations internal operations. Refer to figure 1.1 for a summary of the network layout.

Server 1. *Diego*:

IP Address: 192.168.45.10
Operating System: Debian Linux 2.2r4
Hardware: Customised PC Based -
- Intel Pentium III 900Mhz.
- 256MB RAM.
- 20GB IDE Hard Disk

Primary Role: Webserver for the GIAC Enterprises Fortune Cookie Website. Handles HTTP requests as well as secure HTTPS based requests.
Essential Packages and Versions on System:

Apache Webserver	1.3.9-14
Apache mod-ssl	2.4.10-1.3.9-1
PHP4	4.0.3pl1-0
PHP4-mod-postgres	4.0.3pl1-0
WU-FTPD	2.6.0-5.3
SSH	1.2.3-9.3
Kernel	2.2.20

Server 2. Blade

IP Address: 192.168.45.5
Operating System: Red Hat Linux 7.2
Hardware: PC Based -
- Intel Pentium 4 1.8Mhz.
- 512MB RAM.
- 2X 20GB U/W SCSI Hard Disks

Primary Role: Postgres database server for backend storage of data from the public website running on *diego*.

Essential Packages and Versions on System:

Postgres Database	7.1.3-2

OpenSSH	2.9p2-7
Kernel	2.4.9-34

Server 3. Cortez

IP Address: 192.168.45.8
Operating System: Red Hat Linux 7.2
Hardware: PC Based -
Intel Pentium III 667Mhz.
512MB RAM.
8GB U/W SCSI Hard Disk
Primary Role: DNS Server, housing numerous zone records for the organisation, as well as internal host information. Also the primary mail server for the organisation, specifically running HP Openmail.

Essential Packages and Versions on System:

BIND	9.1.3-4
OpenSSH	2.9p2-7
HP Openmail	7.0
Kernel	2.4.9-34

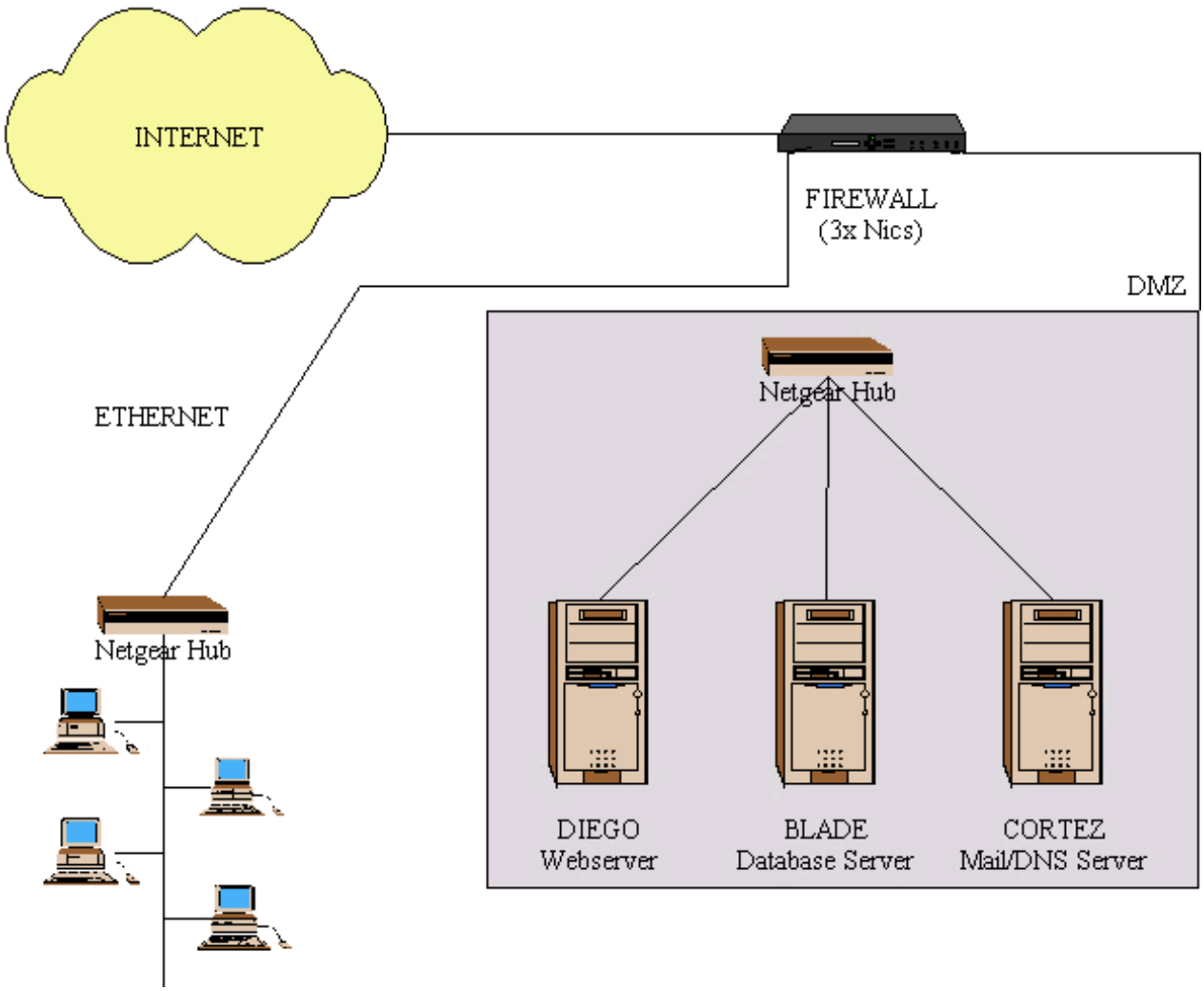

Figure 1.1 – DMZ/Server Infrastructure.

Other Network Hardware:

Firewall: The firewall on the network is a custom built Intel PC running Debian Linux 2.2r6. This version of Debian has been customised with the secure Linux 2.4 kernel module (from the LIDS project - http://www.lids.org/) and IP Tables to provide network based firewalling for the organisation. The firewall also handles NAT for the organisation to get external access to the Internet from the DMZ and corporate ethernet. The firewall also contains VPN software (freeswan) for remote users to access the network using the IPSEC protocol.

Essential Packages and Versions on System:	
OpenSSH	2.9p2-7
Freeswan (Linux IPSEC Implementation)	1.97
Kernel	2.4.18*

* Contains LIDS version 1.1.1r2

Network Connectivity: All server equipment is connected via a Netgear 10/100 Switch. This switch handles connections for the DMZ and firewall. The switch connects to another Netgear switch to provide access from the corporate LAN.

Internet Access: The Internet connection provided by the upstream provider contains 3 public IP addresses, one of these addresses is for the firewall itself. The remaining 2 addresses are translated at the firewall to *diego* and Cortez, to provide inbound website and DNS/mail traffic from the Internet respectively. As mentioned above, the firewall handles all outbound traffic using NAT translation techniques, this allows the DMZ and workstations on the corporate LAN to have Internet access through private IP address allocations.

Network / Infrastructure Recommendations:
The firewall can be improved by implementing application proxy systems that transparently communicate to hosts behind the firewall, this improves security as the remote user is only making connections to the firewall, not the host behind the firewall. This methodology can be found with such applications as the Linux Virtual Server (LVS) project. The benefits of this system also include load balancing and clustering to provide scalability in the future for primarily web and other high demand services.

1.2 Auditing Tools

The auditing tools used in this audit are common applications used to test both the network's permitter security and actual auditing of the local system via terminal access (access provided by the UNIX Administrator). Several of these tools provided similar types

of information, this is mainly to ensure that the perimeter tests match the results found in the local system testing, and vice-versa.

Analysis Programs and Versions:

Program	Version	Switches	Category
LSOF	4.57	-i TCP/UDP:*port*	Internal/Local
Chkrootkit	0.35	-p /mnt/fd0	Internal/Local
John the Ripper	1.6-17	/etc/passwd /etc/shadow	Internal/Local
Nessus	1.1.13	None	Perimeter/Remote
NMAP	v2.54BETA32	-sS -O	ALL

The explanations and purpose of the above programs is categorised below based on local or external (perimeter) usage.

Perimeter/Remote Testing:

The primary tool used to test the perimeter of the network is the NESSUS scanner. This scan was performed outside the network and evaluates 2 main security perspectives. First to test the security measures taken at the firewall for public access to *diego*, secondly to test the system for any vulnerabilities or misconfiguration issues via a local vulnerabilities database (this database is regularly updated by contributors of the Nessus project). Output from the Nessus scan can be found in the appendix II of this document.

Internal/Local Testing.

Local Testing was performed at the console by using several programs provided by the OS, other 3rd party applications were also installed and run specifically for this audit.

Built-in/Default OS Applications:

LSOF – This application is used to audit running processes on a system. This can be for many reasons, foremost of these is to ensure that services running on a particular port are applications that should be running. For instance, if the LSOF scan is performed using the -i TCP:*port* switch, it will display any active TCP connections on the host, and the applications that control that connection. This is useful to ascertain if rootkits or any unnecessary services are running on the host.

TOP – Another OS application, top primarily is used to display the running processes on the system, however top also displays the running hardware usage on the host. Top will display memory usage and CPU utilisation that enables reporting of the server's hardware. This can also be used to find potential software problems, to find which

applications are consuming or draining the system's resources under varied conditions and high load.

3rd Part Applications:

Chkrootkit – This application scans the local file system for default installations of common rootkits installed by hackers or other unauthorised users. The purpose of rootkits is simple, once an attacker has penetrated a system, generally they then install a rootkit that allows a 'back door' into the system so that the user can connect to the system undetected. The chkrootkit program is performed using separate 'detached' binaries of programs required by chkrootkit. A provided floppy disk, set read only, contains the necessary binaries with independent libraries is used to run the scan. This ensures that all tests are performed in a 'clean' environment.

John the Ripper – Passwords are a vital part of keeping the system secure. This application scans the /etc/passwd and /etc/shadow files on the system in an attempt to guess the password used by a user. The auditing team has a custom compiled list of words that we use to find weak passwords, as well as the /etc/dict wordlist preinstalled on the system.

Testing using the NMAP Scanner.

The NMAP network scanner scans a single host or entire subnet for the port status of a host. This information is valuable to see which services are currently and accepting connections on a host. NMAP also has other options, being OS Fingerprinting, NMAP attempts this by 'guessing' the OS based on how the host accepts and handles TCP requests. The importance of the NMAP scan was that the scan was effectively run from 3 locations, externally, from the LAN and from the server itself. (Note that NESSUS actually uses the NMAP program to create a list of ports to scan). This is to test for any open ports that may be denied or rejected through the perimeter firewall, and may not be reject from the LAN.

As mentioned above, some of these tools provide the same information. They allow us to investigate and compare reports from scans that are performed from the local server and remote locations to ensure that they comply with the policies of the network.

Section 2 – Detailed Analysis

This section contains detailed results found from the security analysis performed on *diego*. As mentioned above, *diego* is running Debian Linux 2.2r4. Debian is most known throughout the Linux community for a large focus on OS security, however since the initial release of 2.2r4, there have been many vulnerabilities released that effect several key services and applications on *diego*. This section also outlines patch management, OS configuration, third party application analysis and issues found in administering and maintaining the network, as well as *diego*.

2.1 Operating System Analysis – Vulnerabilities

A critical part of this audit is to identify issues that exist in the operating system and applications running on the server. Debian has release several security advisories (referred to as DSA or Debian Security Advisories) that contain necessary patches to the system to maintain security. The operating system programs contained vulnerabilities as of the 15th of May 2002, the below list is only effected packages that were installed on this system (sorted by date). The impact on the running and security on the server is also summarised below each vulnerability, impact levels are as follows:

CRITICAL:	This is a highly exploitable vulnerability that results in root access on the server.
HIGH:	This level represents vulnerabilities that are highly exploitable and run as the service user.
MEDIUM:	Local Services that are running on the machine, requires console or remote access in order to exploit.
LOW:	Bug or exploit based on package component not in use.

[10 Mar 2002] DSA-120 mod_ssl - buffer overflow
Summary:

> Ed Moyle recently found a buffer overflow in Apache-SSL and mod_ssl. With session caching enabled, mod_ssl will serialize SSL session variables to store them for later use. These variables were stored in a buffer of a fixed size without proper boundary checks.
>
> To exploit the overflow, the server must be configured to require client certificates, and an attacker must obtain a carefully crafted client certificate that has been signed by a Certificate Authority which is trusted by the server. If these conditions are met, it would be possible for an attacker to execute arbitrary code on the server.
>
> This problem has been fixed in version 1.3.9.13-4 of Apache-SSL and version 2.4.10-1.3.9-1potato1 of libapache-mod-ssl for the stable Debian distribution as well as in version 1.3.23.1+1.47-1 of Apache-SSL and version 2.8.7-1 of libapache-mod-ssl for

the testing and unstable distribution of Debian.[1]

Impact on *diego* - Low: As Apache is not configured to use client based certificates (the server only has a single 128Bit server certificate), this vulnerability does not apply to *diego*, however it is still recommended that this package be upgraded as it is a critical application on the server.

[02 Mar 2002] DSA-115 php - broken boundary check and more
Summary:

Stefan Esser, who is also a member of the PHP team, found several flaws in the way PHP handles multi part/form-data POST requests (as described in RFC1867) known as POST fileuploads. Each of the flaws could allow an attacker to execute arbitrary code on the victim's system.

For PHP4 they consist of a broken boundary check and a heap off by one error.

For the stable release of Debian these problems are fixed in version 4.0.3pl1-0potato3 of PHP4.[2]

Impact of *diego* - Critical: This exploit could allow a remote attacker to craft form submissions in URL strings to execute arbitrary code on the system. This will only effectively allow the user to run applications from the webserver user (www-data). This is a major issue as this could allow the user to get access to code on the system that may contain credentials to the database server on blade and other sensitive information. The PHP package should be upgraded immediately.

[18 Feb 2002] DSA-113 ncurses - buffer overflow
Summary:

Several buffer overflows were fixed in the "ncurses" library in November 2000. Unfortunately, one was missed. This can lead to crashes when using ncurses applications in large windows.

The Common Vulnerabilities and Exposures project has assigned the name CAN-2002-0062 to this issue.

This problem has been fixed for the stable release of Debian in version 5.0-6.0potato2. The testing and unstable releases contain ncurses 5.2, which is not affected by this problem.

There are no known exploits for this problem, but we recommend that all users upgrade ncurses immediately.[3]

Impact of *diego* - Low: This is a local exploit as ncurses is a local application. For this to be exploited, the attacker requires access to a terminal (either physically or via ssh) to the server. For this reason, the impact is low, however to prevent further

implications, the package should be upgraded.

[13 Jan 2002] DSA-103 glibc - buffer overflow
Summary:

A buffer overflow has been found in the globbing code for glibc. This is the code which is used to glob patterns for filenames and is commonly used in applications like shells and FTP servers.

This has been fixed in version 2.1.3-20 and we recommend that you upgrade your libc package immediately.[4]

Impact on *diego* – High: Glibc is a core program on Linux systems, almost every application requires the use of glibc (libc) in order to run. Due to the importance of this application, it is recommended that his package be upgraded urgently.

[16 Jan 2002] DSA-102 at - daemon exploit
Summary:

zen-parse found a bug in the current implementation of at which leads into a heap corruption vulnerability which in turn could potentially lead into an exploit of the daemon user.

We recommend that you upgrade your at packages.

Unfortunately, the bugfix from DSA 102-1 wasn't propagated properly due to a packaging bug. While the file parsetime.y was fixed, and yy.tab.c should be generated from it, yy.tab.c from the original source was still used. This has been fixed in DSA-102-2.[5]

Impact on *diego* – Medium: This could allow a local user to exploit the AT daemon and run programs under the daemon user. It is recommended to upgrade the package.

[03 Jan 2002] DSA-097 exim - Uncontrolled program execution
Summary:

Patrice Fournier discovered a bug in all versions of Exim older than Exim 3.34 and Exim 3.952.

This problem has been fixed in Exim version 3.12-10.2 for the stable distribution Debian GNU/Linux 2.2 and 3.33-1.1 for the testing and unstable distribution. [6]

Impact on *diego* - Low: Exim is the main mail server configured on *diego* to handle sending and of mail through the php_mail interface. The configuration in Exim is not

actually applicable to this exploit, however it is recommended that the package by updated.

[05 Dec 2001] DSA-091 ssh - influencing login

Summary:

If the UseLogin feature is enabled in ssh local users could pass environment variables (including variables like LD_PRELOAD) to the login process. This has been fixed by not copying the environment if UseLogin is enabled.

Please note that the default configuration for Debian does not have UseLogin enabled. [7]

Impact on *diego* – Low: This advisory does not effect *diego*, as the UseLogin option in sshd_config is not enabled. However as ssh is the only remote method to connect to the server, it is highly recommended to upgrade the package.

[03 Dec 2001] DSA-087 wu-ftpd - remote root exploit

Summary:

CORE ST reports that an exploit has been found for a bug in the wu-ftpd glob code (this is the code that handles filename wildcard expansion). Any logged in user (including anonymous FTP users) can exploit the bug to gain root privileges on the server.

This has been corrected in version 2.6.0-6 of the wu-ftpd package. [8]

Impact on *diego* – Critical/High: This is a root exploit, allowing an attacker to get root access to the server, this requires immediate attention, as the service is installed to upload files to the webserver. This could allow remote users to exploit and run root applications on the server. As the firewall is blocking FTP connections to *diego*, this would have to be exploited from the local network. It is still recommended to upgrade the package to avoid future security holes.

2.2 Operating System Patch Management

It has been found during the audit that there were no updates performed on *Diego* since it's implementation of Debian 2.2r4, this has also been shown above. The main reason for this is lack of understanding of the Debian OS and the upgrade mechanism's used. The previous UNIX Administrator had implemented the server and did not document or demonstrate to the current UNIX Administrator the methods in maintaining a Debian based system. Fortunately, Debian uses the most efficient method of updating packages known on Linux systems through the Advanced Package Tool (APT). This program updates packages directly from an online archive and installs the packages locally using the Debian Package Manager (DPKG). To initiate this configuration, the system must be configured for the Debian security apt source. To accomplish this, we added the following line to /etc/apt/sources.list

deb http://security.debian.org/ potato/updates main contrib non-free

Once we had entered the above line, we ran the apt-get update command from the console to create a list of updated and new packages that were available from the sources specified in /etc/apt/sources.list. Once this is done, upgrading the system can be accomplished by running the apt-get upgrade command. This searches for updated versions in your local apt list (in /var/cache/apt) and downloads newer packages. The dpkg program then installs the packages based on dependencies.

From discussions with the UNIX Administrator, it was also noted that there was a low concern with system updates since the implementation of the firewall (this was implemented by an external security company). This was due to lack of understanding of the firewall and lack of time for the Administrator to priorities system upgrades.

To overcome these issues, the UNIX administrator needs to subscribe to vendor and security mailing lists to be kept up-to-date with security-related patches and their importance when they are released. These lists will detail the exploit found and often priorities and describe the scope of the exploit. It is highly recommended to join the Debian Security Announce list to be informed of new security releases on *Diego*.

Debian Security Announce List – http://www.debian.org/MailingLists/subscribe
CERT - http://www.cert.org/contact_cert/certmaillist.html

Both these lists will allow the UNIX Administrator to be informed and act on security exploits that are found in applications running on *Diego*.

2.3 Operating System Configuration Analysis.

It has been found during the audit that several configuration issues were found, although these issues are minor, the result in the misconfiguration could result in a system compromise. When setting up a new server, it is recommended to have a checklist of steps to perform when installing packages and configuring the OS in a production environment.

…any large organizations have developed standard installation guidelines for all operating systems and applications used by the organization. These guidelines include installation of only the minimal features needed for the system to function effectively….[9]

Sans top 20 list (http://www.sans.org/top20.htm)

The following list contains OS configuration issues that were found during the audit of *diego*.

- Inetd – The Internet daemon (inetd) was found to be running several non-essential services that were not required. These services included the commonly overlooked

echo, daylight, time and discard services. These services have a history of compromises and are generally only required for network diagnostic purposes and should not be used on a production server. Since none of mission critical programs require inetd, it is recommended that the package be removed.

- OS Banners – During the Nessus scan performed on the server, it was found that numerous services were displaying banner information apon connection that displayed version and other sensitive information. This information could be useful to a remote user doing reconnaissance on the system to ascertain what versions of applications are on the host. Banners were found in Apache and SSH.

- BIND – Bind was installed on the server, the DNS service had been configured for local forwarding so that it could forward local requests to Cortez for lookups required by the system. This method is not required for the server to resolve names, Cortez is the DNS server for the organisation, it is recommended to remove bind from *diego* and specify Cortez as the resolver in /etc/resolv.conf. This is all that is required for the local system and services to resolve DNS requests.

- SSH – The Nessus scan found that the SSHD configuration was allowing version 1 SSH connections, this is not recommended as version 2 has several additional features and better encryption options that version 1. As version 2 is compatible with client connections it is recommended to remove backward compatibility in /etc/ssh/sshd_config.

These configuration mistakes would be avoided if the organisation created a server implementation guideline as specified above, this would contain step by step instructions on hardening and securing the OS installation.

There were other applications on the server that had been removed. Several high-risk security packages were removed from the system that are installed by default, these packages included the 'r' suite of programs (rlogon etc), telnetd, fingerd, NIS and NFS. These applications were not required for the running and administration of *Diego* and were therefore not installed on the system.

2.4 Installed Third Party Software

Apache - running the companies primary website, has been configured using Debian's standard .deb packaging system. There were a few options in the apache configuration that should be reviewed for security. The main option is that apache is allowing directory listings, that is when a directory on the webserver does not contain an index file, the whole directory is read. This allows anyone to save server scripts and website configuration files (eg database credentials), although there were no directories on in the /var/www/ directory that didn't contain an index file, it is good practice to remove deny directory listings for the reasons stated above. Other configuration options included cgi

access. The site is primarily written in PHP4, and it has been found that there were cgi files in the cgi-bin directory that had been obsoleted and not removed from the directory, some of these scripts allowed mail to be relayed through the local mail process. This could result in a remote user spamming from the cgi mail script.

Mail server (Exim) – Due to the website requiring email facilitates, the UNIX administrator and developers agreed that it was necessary to install a mail server on the system in order for the website to send email to website customers. However the mail server program, Exim, had not been configured for anti-relaying functionality. The Exim configuration file (/etc/exim.conf) was set to relay from any network address. Although the firewall was restricting smtp access to the server, it is once again good practice to setup the mail server so that mail can only be sent from localhost (as the organisations mail server handles staff emailing).

WU-FTPD – The FTP server was implemented to assist developers in uploading new scripts for the website on *diego*. The WU-FTPD is the default FTP daemon in Debian Linux. The server has been configured to only allow logins by local users, anonymous access is not allowed. All documents downloaded by remote users/clients are all performed via the HTTP protocol, FTP is only required for developers to upload files. There is one security issue as mentioned above with the WU-FTPD services. Several other solutions are available to overcome some of the security issues found in FTP. This includes the secure copy (SCP) system as part of the SSH package, refer to the recommendations section 3 for more information.

2.5 Administrative Practices

Apart from the auditing of applications and security measures on *diego*, it is also important to ensure that these measures are being responded to efficiently and effectively. Below details analysis found in the overall network operations of the organisation.

Staff Summary:

GIAC's IT department consists of 4 staff members, that varies in their specific roles:

1. IT Manager, responsible for the network infrastructure and IT staff, reporting issues and needs to upper management. The IT manager is knowledgable in UNIX and handles excessive workload when issues occur on the DMZ.

2. UNIX staff member responsible for installation, maintenance and support of the DMZ Linux servers.

3. Staff member for internal support to 20 staff members in the organisation, including

upper management. Assisted by external Support Company.

4. PHP/database developer that assists external developers in producing and maintaining the website.

Maintenance:

The daily maintenance of the server consists of UNIX administrator manually checking the syslog, maillog and apache logs of the host. This process can take large amounts of time for an already overworked IT department.

To overcome this loss in productivity, the IT department can use several open source applications that manage server logs and resources in a centralised location. Logcheck is an application that can be highly customised to periodically send emails to the UNIX or IT Administrators when issues arise on the server. Logcheck is configured via several text files that specify typical messages found in logs and 'violations' that are found in logs. Logcheck compiles a report periodically (can be configured through cron) based on the rules in the text files.

Host/Service Monitoring:

The IT department does not have a network wide monitoring service, several applications are available to monitor the health and critical services of servers on the network. Netsaint, a popular GPL application, is a perl-based host and service monitoring tool. Implementing this software will allow the UNIX Administrator to configure all hosts on the network and monitor the services on these hosts. For example the https (443) and http (80) protocols on *diego* would be monitored, as well as ssh (23) to ensure that the remote access protocol is running at all times. Netsaint will also notify administrators of the host's connectivity from the monitoring machine, if the machine is under heavy load or congestion, the monitoring service can report on the host's status in these conditions. The program is flexible in reporting faults, either by email, pager or sms. These faults can be escalated based on groups, for example, first instance issues can be reported to the UNIX Administrator, then if the problem is not resolved, it is then escalated to the IT Manager. All functions of the server are configured and maintained via a web interface on a central 'monitoring' server. This server can be a host within the organisation's local network.

Issue Management:

Keeping track of issues that occur on servers can often be a difficult procedure, especially if there is only one Administrator to support all the servers. For this reason, an issue management system will allow the administrator to track issues that have occurred on the

server and prioritise and act on these issues while documenting information to a single location. The issue management will also keep histories of issues and resolutions to provide valuable information when troubleshooting future issues. It is recommended to implement an issue ticketing system for Administrator and Manager to follow, such an application will allow any IT staff members to manage issues and tasks for the servers from a centralised location. Procedures can be created for IT staff on logging issues found on servers and document steps taken to resolve the issue(s).

Intrusion Detection Systems (IDS):

The network does not run an IDS, these applications are essential in tracking remote attacks and other suspicious activities performed by remote users. The IDS system would be installed on the firewall and set to monitor all traffic entering the network from the Internet Service Provider, this would allow the UNIX Administrator to view reports of suspicious traffic and other incidents that have occurred on the network. An example is the SNORT project, which offers full network based IDS, including several methods that can be implemented by the IT Department to report on issues found by the IDS.

2.6 Protection of Sensitive Data

Sensitive data on system:

Remote access: The server is being accessed via the SSH protocol. The firewall is allowing remote SSH connections to the system. This is for the purpose of allowing the UNIX Administrator 24x7 access to *Diego* and other servers on the DMZ. This should be reviewed as the firewall is already running the IPSEC implementation, this should be enabled for the UNIX Administrator to connect the DMZ via VPN, this will further secure the SSH connectivity from remote locations.

SSL Certificate (mod_ssl): The SSL certificate ssl.giac.com is a certified certificate. The certificate files are located in /etc/apache/certs. The private key has been found to have world-readable access, this is a major flaw, which could allow a local attacker to view the private key on the system and manipulate the SSL certificate. It is recommended that the private key be placed in a secure area that only has read access by root, as this is all that is required for the webserver to read the key on the starting of the httpd service.

File Integrity: There is no file integrity checking system installed on *Diego*, this will seriously jeopardise any attempt to trace changed files on a file system in the event of a system compromise. Installation of a integrity database, such as Tripwire, will provided detailed information about changes made to files in the filesystem. This is vital to ascertain if files have been modified or tampered with. It is recommended that the Tripwire system be installed to monitor file changes on *diego*.

Sensitive Data in transit:

The information that is being sent from the webserver to the database server is not being encrypted, this allows the traffic to be sniffed by an attacker and therefore intercept SQL commands that contain insertion and retrieval of data from the database. The likelihood of this method being executed is low, as the database server is highly secured and access to the database itself is only allowed from the webserver. However to reduce the possibility of sniffing data in transit, it is recommended to install a secure tunnel between the 2 hosts. This can be accomplished in a variety of methods, stunnel is a simple method of encrypting data between two locations based on services. This would allow postgres information (running on port 5432) to be encrypted when being transmitted from then webserver to the database on blade.

2.7 Access Controls

As *Diego* is remotely administrated via the SSH protocol, this is the recommended access type as all data transit is encrypted from the client to server. However it was found during the audit that the SSH server had been configured to permit root access, via the permitrootaccess option in /etc/ssh/sshd_config. This should be removed so that the Administrator and developers (if necessary) have to login under their normal account and use the 'sh' command to obtain root access, or alternatively SUDO can be configured (see below). There is only access to the physical console from the IT manager and UNIX Administrators who have access to the server facility at GIAC. Both of these staff members are available in the event of a emergency access that may be required.

Several unnecessary users had access to the root account, this included developers who required one-off access to services under root's ownership. To maintain and audit the commands performed by system administrators and developers requiring root access, the SUDO program should be implemented and used by staff to customise access to root commands on a user-by-user basis. This will allow an audit trail of users running root commands.

Diego was scanned for weak passwords using the John The Ripper password-cracking program to find any user accounts on the system that contained easy or weak passwords. The program found several passwords on the host that had simple English based passwords. It was also found while reviewing the /etc/passwd and /etc/shadow files that there were several user accounts still on the system for contact or terminated employees. Several logins for the system also contained shell access, this allowed owners of programs to interactivity login to the command shell and run applications in the environment, including shutting down of the service owned by the user.

Ownership of files was evaluated. After the completion of several tests made in key directories of the server (mainly /home /root /etc /boot and /var) it had been found that the root's home directory was world readable. This is setup by default on Debian systems and should be set to no world readable access, this ensures that files created in root's home directory are only readable by root. This also applies to user's home directories, as with root, Debian creates user directories world readable, this should be changed to

maintain a tighter security for users on the system. There were several world writable files found on the system, several of these were in the webserver directories. This is dangerous as it allowed remote users to upload files into the directory without any authentication or ACL setup.

2.8 Backup and Disaster Recovery Strategies

The backup system implemented in *diego* is sufficient in the event of data corruption or accidental deletion of data on the system. This is accomplished via regular tape backups that are made to a 20GB DAT backup unit that is installed on the system. The backup rotation is performed daily, and every 2nd tape is transported offsite, as well as the weekly tape created on Friday's.

There was a lacking in procedure for disaster recovery, this is in the event of critical hardware failure that could result in both software corruption and hardware replacement. Due to the importance of *Diego*, several options are available to prevent hardware-based failure.

1. Redundant Hardware on system: *Diego* could be implemented with RAID 2, this allows mirroring of the local disks, in the event of a hard disk crash or failure, the mirroring disk can continue to operate the server while the faulty hard disk is replaced. Similar resource duplication can also be applied to the power supply. The power supply and hard disk are often the most common component to fail on a server.
2. Duplicated System: This is the recommended option for maximum disaster recovery. This process involves purchasing a duplicate server to replace *diego* in the event of a critical failure. This method is encouraged as the server can also act as a development environment to the website and application development.

Either one of these options will better equip your IT team in the event of a system failure. Policies should be implemented to test the backup procedures by monthly or biannually restoration of backups to ensure they are verified and recoverable.

Section 3 - Priority Issues and Recommendations:

Apon completion of the audit, the auditing team has documented several issues that require immediate attention in order to resolve and secure the server. This section will also address additional recommendations that were mentioned in this report that will increase productivity for the IT department as well as provide secure connectivity and audit trailing for future audits and the UNIX Administrator.

3.1 Priority Issues

These issues all need immediate attention, the list below priorities the issues based on urgency and potential security threat. Approximate time to resolve each issue is also included to allow the UNIX Administrator to allocate time to implement these recommendations and issues.

1. Application Updates:
Foremost of these issues is to update the server, this is critical to prevent security breaches on the server. As mentioned in the report, this can be accomplished with the two following commands:

apt-get update: This command will retrieve a list of all the updated packages on the system and versions, it then compares the new downloaded package list with the currently installed packages on the system. In order for this function to work, the apt package must be configured to connect to http://security.debian.org in order to retrieve the latest patches. Put the following line into the /etc/apt/sources.list file to get security updates:

deb http://security.debian.org/potato/updates main contrib non-free

Apt-get install: This command will list any packages that require updating due to bugs or security issues. The command will then download and automatically install the packages onto the system using the dpkg program.

Approximate Time involved:	6 Hours. (This includes full testing of the packages to ensure they do not effect other applications on the system)

2. Password/User Management
Reset passwords, remove unnecessary user accounts. Deigo contained many user accounts on the system that were no longer required as they were used by external developers and former employees, one of these being the previous administrator. It is recommended to delete the user accounts no longer active and change the root password. Passwords should be at least 8 characters in length and contain at least two non-alphanumeric characters. John The Ripper also reported that several passwords were easily detected by the system. These passwords should be reset using the

password convention mentioned above.
The root access account should be changed every 4-8 weeks to maintain maximum security on the system. These password policies can also be applied to the other servers on the DMZ

Accounts found with bad passwords include:

1. jack
2. nick (Former administrator)
3. adam
4. donna

Approximate Time involved: 5 Hours.

3. Configuration Changes
Resolve configuration errors found on the server. This step includes removing the unnecessary packages installed on the system during initial setup, this includes inetd and bind. This also involves the reconfiguration of Exim and apache to harden the server.

4. Remove inetd
   The inetd package is running several non-essential services, since these services are not required, it is recommended that the inetd package be removed.
   This is done using dpkg with the command:

   #dpkg –purge inetd.

5. Remove bind
   The bind package is installed on the system when it is not required, for better overall security, remove the bind package and update /etc/resolve.conf to search Cortez in order to perform DNS lookups. To remove the package, issue the following command.

   #dpkg –purge bind.

6. Remove OS banners off applications. This requires modify the configuration files of SSH, apache and WS-FTP to remove the application versions on the server.

Approximate Time involved: 5 Hours

7. Resolve world-write / world-read file issues.
There are several files and directories that contain bad permission's, either sensitive files containing world-read access or directories that are world writable. This requires urgent attention as the sensitive information and directory information can be read by unauthorised users.

Recommendations:

1. Remove or change permission's of world-writable files under the /var/www directory.
2. Move the private SSL key from /etc/apache/certs to /etc/apache/private and set the directory and all files in directory to be owned and read only by root. It is also unknown if the key has been read, it is also recommended to revoke and renew the key with the authorised certificate body to obtain a new private key.

Approximate Time involved: 8 Hours

8. Apache HTTPD Configuration.
This process involves removing directory access and strengthening security of the CGI path on the server. This would deny access to directory listings on directories that do not contain index files. This process also includes working with developers to remove unnecessary scripts in /cgi-bin on the webserver as found by the Nessus scanner.

Approximate Time involved: 5 Hours

9. Remove SSHD Backwards compatibility
The SSH daemon is allowing backward compatibility to the version 1 protocol. This is known to cause major security issues, as the access to the server is being done using clients that are able to communicate using version 2 of the SSH protocol. It is recommended to remove the version 1 support. SSHD is also configured to allow root logins, this should also be set to no. This will prevent unauthorised logins attempting to crack the root password by brute force.

Update /etc/ssh/sshd_config with the following line and restart the daemon.

Protocol 2
Permitrootlogin No

Approximate Time involved: 2 Hours

10. Install Tripwire File Integrity System
As mentioned in the report, there is no file integrity checker or database on the system, this is a vital component when a system has been compromised, it is the only method in verifying file changes in a filesystem. Tripwire runs on the system and tracks changes made to the files in the filesystem and notes these changes in a database, these changes are then sent to the UNIX Administrator for review. Tripwire should be setup on a separate partition and permissions should be hardened to prevent tampering of the

database data.

Approximate Time involved: 4 Hours

11. Remove relaying config from Exim.

After performing the NMAP on *Diego* and determining that there was a mail server running, it was found that the mail server, Exim, had been configured to accept mail from any host. This configuration could potentially allow spammers to use the mail server to relay mail. The firewall was dropping smtp connections to *Diego*, so external abuse could not be made, however local abuse was possible. Since the mail is only required by the local webserver, there are 2 methods to resolve this issue.

1. Configure Exim to only relay local mail.
This is done by adding *host_accept_relay = 127.0.0.1* to the /etc/exim.conf file.

2. Re-configure the web scripts that use email to connect to Cortez to send the mail and remove Exim. (This is based on the website script capabilities)

Approximate Time Involved: 4 Hours

3.2 Other Recommendations:

The following recommendations are based on the audit performed. These effect both changes to the server, as well as management and running of the entire network.

Access Levels

Implement SUDO for developer access to root privileges. This process involves the implementation and configuration of SUDO for users to gain access to root applications and services.

The process would be configured for developers to interact with the system as follows. Developers would have group access to the /var/www directory and SUDO ability to start/stop web services on *Diego*. This is accomplished by adding users and commands to the SUDOers file in /etc.

The 'root' password should only be allocated to 2 or 3 individuals in the organisation, these individuals would include at least the IT manager and UNIX Administrator, any other user requiring root access should be setup using SUDO.

Approximate Time involved: 6 Hours

Backup/Disaster Recovery

Implement a new server on the network to provide hardware backups and testing and evaluation of applications and developer code in a development environment that will not

effect the stability of the production server. This server must be of similar hardware for the disaster recovery of *Diego* to be effective.

Approximate Time involved: 10 Hours.

System Management - Logcheck

Logcheck is highly recommended for monitoring *diego*. Every hour Logcheck can send a summary from the logs on *diego*, note that Logcheck will only send a message if there is an issue, this saves administration time.

Approximate Time involved: 3 Hours

Issue Management

Several programs are available to keep track of issues on servers for administration staff, recommended packages to implement include request tracker (Rt) which can be found at http://www.fsck.com/projects/rt/. This program allows the IT department to log and action issues that occur on servers (and even workstations) at GIAC. This system will be run from a web interface and interact with a postgres or mysql database running on blade. The web interface is protected by a username and password and contains access groups for tailored access rights to the system.

Approximate Time involved: 6 Hours (includes 1 hour training)

Network Host/Service Management

The implementation of Netsaint consists of installing the monitoring component on a Unix webserver. This machine can be installed on the local ethernet or on the DMZ, the webserver is then configured with Netsaint and apache for the monitoring interface, once installed, the Netsaint service is configured with the essential hosts on the DMZ, including the firewall. All services are then created in the Netsaint configuration for monitoring, the basic monitoring is as follows:

Diego	***Cortez***	***Blade***
SSH (TCP 21)	SMTP (TCP 25)	Postgres (TCP 5432)
HTTP (TCP 80)	POP3 (TCP 110)	
HTTPS (TCP 443)	IMAP4 (TCP 139)	
	SSH (TCP 21)	
	DNS (TCP/UDP 52)	

The services will then be configured to email system Admins when the hosts or services are down or under heavy load.

Approximate Time involved: 6 Hours

Intrusion Detection System (IDS) Implementation

This involves the implementation of the SNORT IDS on the firewall, this will allow the IT Manager or UNIX Administrator to monitor the network for suspicious traffic on the network, as well as alerting on commonly exploitable code and gateway activity.

ACID will be implemented on a webserver to monitor and record (via a postgres database on blade) the activity logged by the snort daemon.

Approximate Time involved: 6 Hours

Secure FTP (SFTP) Implementation

Due to the several issues found with the FTP server implementation for *diego*, it is recommended to look at other methods of sending files to the host when updating new website changes. SCP, which is part of the SSH protocol, can be implemented to allow users to upload files from clients. Since internal developers run under the Windows environment, a 3rd party application is required to accomplish this, PuTTY is a ssh suite for Windows that allows users to connect via ssh/scp or sftp from a windows client. There are several GUI frontends for this suite that work as FTP clients.

Approximate Time involved: 4 Hours

Sensitive Data Transport

Implementing secure tunnel between webserver and *diego* for database transactions. This can be accomplished using several encryption methods, the simplest is to use stunnel to create a secure tunnel specifically for postgres traffic running on port 5432.

Approximate Time involved: 5 Hours.

Appendix A: References

1. ”Debian GNU/Linux 2.2r4 released”. URL : http://www.debian.org/News/2001/20011105 (1st Jun. 2002)

2. “Security Information”. URL http://www.debian.org/security/ (1st Jun. 2002)

3. “Security Alerts from 2001”. URL: http://www.debian.org/security/2001/ (1st Jun. 2002)

4. “Security Alerts from 2002”. URL http://www.debian.org/security/2002/ (1st Jun. 2002)

5. “Nessus”. 13th Jun 2002. URL http://www.nessus.org (9th Jun 2002)

6. Galstad, Ethan. “Netsaint Network Monitor”. URL http://www.netsaint.org (9th Jun 2002)

7. “The Linux Virtual Server Project”. URL http://www.linuxvirtualserver.org/ (9th Jun 2002)

8. Ruiu, Dragos “SNORT FAQ” URL http://www.snort.org/docs/faq.html (15th Jun 2002)

9. Danyliw, Roman “ACID: Installation and Configuration”, 18th Nov 2001 URL: http://www.andrew.cmu.edu/~rdanyliw/snort/acid_config.html (15th Jun 2002)

10. “PuTTY: A Free Win32 Telnet/SSH Client”. 19th Jun 2002. URL http://www.chiark.greenend.org.uk/~sgtatham/PuTTY/ (22nd Jun 2002)

11. Dioso, Alexander “Using Stunnel with PostgreSQL HOWTO” 29th May 2002 URL: http://cfm.gs.washington.edu/~adioso/HOWTO/StunnelPostgreSQL.xml (22nd Jun 2002)

12. “Request Tracker” URL: http://www.fsck.com/projects/rt/ (22nd Jun 2002)

13. “Tripwire FAQ” URL: http://www.tripwire.org/ganda/faq.php (22nd Jun 2002)

Appendix II – Nessus Output

This outlines output relevant to this report:

List of open ports:

- *general/tcp (Security notes found)*
- *domain (53/tcp) (Security warnings found)*
- *www (80/tcp) (Security notes found)*
- *ssh (22/tcp) (Security warnings found)*

Information found on port general/tcp

Nmap found that this host is running Linux Kernel 2.2.20 (X86)

Warning found on port domain (53/tcp)

The remote name server allows recursive queries to be performed by the host running nessusd. If this is your internal nameserver, then forget this warning. If you are probing a remote nameserver, then it allows anyone to use it to resolve third parties names (such as www.nessus.org). This allows hackers to do cache poisoning attacks against this nameserver.

Solution: Restrict recursive queries to the hosts that should, use this nameserver (such as those of the LAN connected to it). If you are using bind 8, you can do this by using the instruction, 'allow-recursion' in the 'options' section of your named.conf. If you are using another name server, consult its documentation.
Risk factor: Serious

Information found on port domain (53/tcp)

The remote bind version is: 8.2.3-REL-NOESW

Information found on port www (80/tcp)

The remote web server type is:

Apache/1.3.9-14 (Unix) Debian GNU/Linux PHP/4.0.3pl1 mod-ssl/2.4.10.

We recommend that you configure your web server to return
bogus versions in order to not leak information

Information found on port www (80/tcp)

For your information, here is the list of CGIs
that are used by the remote host, as well as their arguments:

Syntax: cginame (arguments [default value])

/search.php (query topic author days [0])
/index.php (menu [4])
/'user.php (‘)

/FormMail.cgi (‘)

Warning found on port ssh (22/tcp)

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

Warning found on port ssh (22/tcp)

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

Information found on port ssh (22/tcp)

Remote SSH version: SSH-1.2.3

Information found on port ssh (22/tcp)

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

. 1.33
. 1.5
. 1.99
. 2.0

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

Appendix III – Nmap Output

(All scans were using the same version to maintain consistency)

Generated from Localhost (using nmap –sS –O 127.0.0.1)

```
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on 127.0.0.1:
(The 1533 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
22/tcp     open        ssh
37/tcp     open        time
80/tcp     open        http
443/tcp    open        https


Remote operating system guess: Linux Kernel 2.2.20 (X86)
Uptime 12.333 days (since Wed May 1 08:27:17 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 13
seconds
```

Generated from Local Office Ethernet (using nmap –sS –O *diego*)

```
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on diego (xxx.xxx.xx.xxx):
(The 1533 ports scanned but not shown below are in state: closed)
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
22/tcp     open        ssh
37/tcp     open        time
80/tcp     open        http
443/tcp    open        https


Remote operating system guess: Linux Kernel 2.2.20 (X86)
Uptime 12.333 days (since Wed May 1 08:27:17 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 54
seconds
```

Generated from Remote Location (using nmap –sS –O xxx.xxx.xx.xxx)

```
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on xxx.xxx.xx.xxx:
(The 1533 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
53/tcp     open        domain
80/tcp     open        http
443/tcp    open        https

Remote operating system guess: Linux Kernel 2.2.20 (X86)
Uptime 12.333 days (since Wed May 1 08:27:17 2002)

Nmap run completed -- 1 IP address (1 host up) scanned in 22
seconds
```

Appendix IV – Chkrootkit output

Chkrootkit run on separate floppy disk with static binaries required by the system also on floppy. Command Line parameter chkrootkit -p /mnt/fd0

```
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not infected
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not infected
Checking `inetdconf'... not infected
Checking `identd'... not found
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not infected
Checking `mingetty'... not found
Checking `netstat'... not infected
Checking `named'... not infected
Checking `passwd'... not infected
Checking `pidof'... not infected
Checking `pop2'... not found
Checking `pop3'... not found
Checking `ps'... not infected
Checking `pstree'... not infected
Checking `rpcinfo'... not infected
Checking `rlogind'... not found
Checking `rshd'... not found
Checking `slogin'... not infected
Checking `sendmail'... not infected
Checking `sshd'... not infected
Checking `syslogd'... not infected
```

```
Checking `tar'... not infected
Checking `tcpd'... not infected
Checking `top'... not infected
Checking `telnetd'... not found
Checking `timed'... not found
Checking `traceroute'... not infected
Checking `write'... not infected
Checking `aliens'... no suspect files
Searching for sniffer's logs, it may take a while...
Searching for HiDrootkit's default dir... nothing found
Searching for t0rn's default files and dirs... nothing found
Searching for t0rn's v8 defaults... nothing found
Searching for Lion Worm default files and dirs... nothing found
Searching for RSHA's default files and dir... nothing found
Searching for RH-Sharpe's default files... nothing found
Searching for Ambient's rootkit (ark) default files and dirs...
nothing found
Searching for suspicious files and dirs, it may take a while...
nothing found
Searching for LPD Worm files and dirs... nothing found
Searching for Ramen Worm files and dirs... nothing found
Searching for Maniac files and dirs... nothing found
Searching for RK17 files and dirs... nothing found
Searching for Ducoci rootkit... nothing found
Searching for Adore Worm... nothing found
Searching for ShitC Worm... nothing found
Searching for Omega Worm... nothing found
Searching for Sadmind/IIS Worm... nothing found
Searching for MonKit... nothing found
Searching for anomalies in shell history files...nothing found
Checking `asp'... not infected
Checking `bindshell'... not infected
Checking `lkm'... nothing detected
Checking `rexedcs'... not found
Checking `sniffer'...  eth1 is not promisc
Checking `wted'... nothing deleted
Checking `z2'...
nothing deleted
```

Apendix VII -- Endnotes:

---

[1] "Debian GNU/Linux – Security Information" – DSA –120-1 mod_ssl URL: http://www.debian.org/security/2002/dsa-120
[2] "Debian GNU/Linux – Security Information" – DSA –115-1 php URL: http://www.debian.org/security/2002/dsa-115

[3] "Debian GNU/Linux – Security Information" – DSA –113-1 ncurses URL: http://www.debian.org/security/2002/dsa-113

[4] "Debian GNU/Linux – Security Information" – DSA –103-1 glibc URL: http://www.debian.org/security/2002/dsa-103
[5] "Debian GNU/Linux – Security Information" – DSA –102-1 at URL: http://www.debian.org/security/2002/dsa-102

[6] "Debian GNU/Linux – Security Information" – DSA –097-1 exim URL: http://www.debian.org/security/2002/dsa-097

[7] "Debian GNU/Linux – Security Information" – DSA –091-1 ssh URL: http://www.debian.org/security/2001/dsa-091

[8] "Debian GNU/Linux – Security Information" – DSA –087-1 wu-ftpd URL: http://www.debian.org/security/2001/dsa-087

[9] "The Twenty Most Critical Internet Security Vulnerabilities (Updated)" URL: http://www.sans.org/top20.htm