GCUX Practical v1.9 Option 2

# Security Audit of GIAC Enterprises giacmain

Robert A. Napier
4 April 2003

# Table of Contents

# Table of Findings

# 1 Executive Summary

In March, 2003, Robert Napier performed a security audit of GIAC Enterprises' main internal server, giacmain. This audit consisted of phone interviews, physical examination of the premises, external ("black-box") technical investigations and on-host privileged technical investigations.

The most significant finding was an existing root exploit in a user directory. This was clearly an intentional attack on the system installed at least five years ago. It is not clear whether this exploit was currently in use at the time of the audit. The administrators of giacmain were immediately notified, prior to the release of this audit, and corrective measures have already been taken to close the hole. The administrators' investigation is ongoing at this time.

Other critical findings include:

- Many weak user passwords, increasing the likelihood of password-guessing attacks;
- No comprehensive patching strategy, greatly increasing the likelihood that a well-known, exploitable system error ("bug") will be discovered by an attacker;
- No regular backups or disaster-recovery scheme, greatly increasing the cost of an attack or unintentional damage.

These findings should be addressed immediately. Other significant findings that should be addressed in the near term are summarized in Section 6 beginning on page 32. All findings are detailed throughout sections 3, 4 and 5.

Many of the significant security issues on giacmain are policy-related rather than technical. For example, the administrators have generally kept patches reasonably up to date, despite the lack of a clear policy on patching. Many of the technical issues make attacks more likely or more damaging rather than being specific security holes themselves. The administrators appear to have a clear security interest in their dealings with new services, so better security policies will likely be well-received and quickly implemented.

The current security posture of giacmain is insufficient given the importance of the server. Some relatively simple improvements will go a long way towards improving the security posture at minimal cost.

# 2 Overview

## *2.1 Report layout*

This audit report consists of five sections:

- An Executive Summary providing high-level results of the audit suitable for executive management;
- An Overview introducing the client and system to be audited, and describing the audit methodologies;
- A detailed analysis of the audit at the Physical, Operating System and Application layers, providing technical details suitable for the administrators of the system;
- A list of Critical Issues and Recommendations that should be immediately investigated by the administrators of the system; and
- Supporting documentation in Appendices and References.

It is strongly recommended that all owners of the system read the Executive Summary and Critical Issues and Recommendations section at a minimum.

## *2.2 Client*

The client for this audit is GIAC Enterprises, a 121-person e-business specializing in the online trade in fortune cookie sayings. GIAC Enterprises is located at a single site in Raleigh, NC and has been operating for 17 years. They entered the online business eight years ago to expand on their existing “fortune cookie saying slip” (FCSS) manufacturing business. They discontinued their manufacturing business three years ago and subsequently have been an entirely online company.

GIAC Enterprises’ primary asset is their 1.2 million fortune cookie saying database (FortuneDB).

## *2.3 Description of System*

### 2.3.1 Overview

The audited system, giacmain, provides most of the internal services for GIAC Enterprises including home directory storage, email, internal web services and UNIX shell access.

All GIAC Enterprises users have home directory space on giacmain. This is the primary storage location for documents. Users can share documents in various project directories protected by group permissions. Home and project directories are not exported via NFS or SMB, though pressure is growing to provide this access.

All GIAC Enterprises users receive email on giacmain. Users read their email locally using mutt, pine, elm and similar UNIX tools, or forward their mail using `.forward` files.

Most GIAC Enterprises internal web pages are stored on giacmain. These include internal corporate information, group websites and individual (including personal) websites. The FortuneDB fortune cookie saying database is not stored on giacmain, nor is the web-based administration interface.

Users regularly log directly into giacmain to access various UNIX tools (mailers, news readers, web browsers, etc). Giacmain is not directly accessible from the internet, though VPN access is available. VPN services are provided by external hardware and are not covered by this audit.

### 2.3.2 Hardware and OS

Giacmain is a dual-processor Sun Ultra 2 running Solaris 8, with two internal disks and two disk trays containing a total of 20 disks and 158G of disk space.

### 2.3.3 Software

Giacmain provides numerous services including DNS, FTP, HTTP, SSH, mSQL and SMTP. Furthermore, giacmain provides dozens of user-level programs such as IRC clients, compilers, mail readers, etc.

### 2.3.4 Location

Giacmain is located in the GIAC Enterprises data center.

## *2.4 Audit Methodologies*

### 2.4.1 Interviews

Before coming onsite, we performed questionnaires and phone interviews with the following individuals:

- Robert Allen – Lead system administrator
- Mark Bandel – System administrator
- Lorrie Jansen – Fortune cookie saying developer
- Fred Baker – Sales

The intent of our interviews was to determine how giacmain is administered and how various user groups interact with it. The questionnaires assessed level of agreement or disagreement with various statements about the administration, security and usage of giacmain. The phone interviews consisted of open-ended questions following up on the questionnaires and tracking consistency of answers between respondents.

### 2.4.2 Site Investigation

We visited the GIAC Enterprises data center to survey the physical environment and security of giacmain. All access was escorted by GIAC Enterprises personnel.

### 2.4.3 Black-box Investigation

While on site we performed a “black-box” non-intrusive audit of giacmain. This audit included port scans and vulnerability scans, and represents information that can be gathered by persons without an account on giacmain. We did not use scans that could cause denial of service or damage to of any of GIAC Enterprises’ systems.

### 2.4.4 On-System Audit

Also while on site, we performed an on-system, non-intrusive audit with temporary root access via sudo. This consisted of system searches, configuration file examination and similar information gathering activities.

# 3 Physical Security Detailed Analysis

We visited the GIAC Enterprises data center during normal working hours, escorted by a giacmain administrator.

Physical access controls on giacmain are extremely strong, requiring a combination of badge and PIN access to enter the data center. The data center is also monitored by closed-circuit camera 24-hours a day. Door, lock and wall construction of the data center are solid, with no windows. Giacmain is secured in a key-locked rack. The keys are standard duplication-resistant models, one controlled by the data center operations on-call staff member and the other by the manager of Loss Prevention. We did not investigate the fire suppression system.

A stronger protection for the rack would be an electronic lock requiring both badge and PIN and authorizing only individuals who require physical access to giacmain. However, this system can be extremely expensive (in the thousands of dollars) so is likely unwarranted given the strong overall security of the data center and the moderate value of the information on giacmain.

# 4 Operating System Detailed Analysis

## 4.1 Operating system vulnerabilities

Solaris 8 is a well-vetted operating system with timely patches provided by the vendor, an active user community and extensive security documentation. Due to its popularity, attackers are likely also to know it well, and so it makes a common target for new attacks. In general, however, Solaris 8 is an excellent choice for general purpose systems both internally and externally facing.

Administers of Solaris systems generally have to decide whether to rely on Sun's packaged versions of services such as FTP, sendmail, etc., or to build these services from original source. There are tradeoffs for either of these solutions. Using Sun's packages ensures a higher level of support from Sun and is generally easier to maintain. Building packages from sources allows greater flexibility and allows deployment of security patches faster than is possible when relying on vendors to generate packages.

The administrators of giacmain have built most services from original source rather than relying on Sun's packages. By doing this, they have been able to make several systems more secure than Sun's default packaging. This has increased the complexity of maintaining the system, however, particularly when Sun's service cannot be removed conveniently. This is most evident with BIND, which is part of Solaris' core package (`SUNWcsu`). Ensuring that the correct version of BIND (`/usr/local/sbin/named`) runs at startup is error-prone, particularly when OS patches are applied. `/etc/rc2.d/S72inetsvc` has been made into a symbolic link to `local.S72inetsvc` and has been modified to remove the `/usr/sbin/in.named` startup, but this has not been documented.

| | |
|---|---|
| **Finding 1:** | **There are a large number of unused packages. (Moderate)** |
| **Solution:** | **The administrators should go through the entire list of packages and remove any that are not necessary for the functioning of giacmain. Local services that are not used but cannot be removed due to packaging issues need to be documented so that they can be re-checked after patch installations.** |

## 4.2 Security patch installation/management

Security and other patches are applied sporadically on giacmain. As security issues come to the attention of the administration staff, specific patches or upgrades are performed. There is no comprehensive strategy to patching.

| | |
|---|---|
| **Finding 2:** | **There is no comprehensive strategy to patching. (High)** |
| **Solution:** | **Recommended and Security patches should be applied regularly from SunSolve[1]. Due to the large number of source-built packages on giacmain, the administrative staff should also subscribe to the security lists from Sun, CERT[2] and the SANS alert consensus newsletter[3]. Furthermore, they should maintain a list of all source-built packages to compare against announcements from these lists.** |

## *4.3 Vulnerability Scan*

In order to find well-known, externally visible vulnerabilities, we ran the Nessus[4] vulnerability scanner version 2.0.1. We ran all non-destructive tests. A full report is available in section 7.2. Significant vulnerabilities are noted in the relevant sections of this report.

## *4.4 Configuration vulnerabilities*

Some excellent public sources of configuration information for Solaris are the Solaris Security FAQ[5], Defense In-Depth on a Solaris 2.X System[6] and *Solaris Security Step by Step*[7]. We began the configuration audit of giacmain against these resources.

### 4.4.1 root configuration

The root user should have a umask of 027 or 077 to disallow world access. root should also have a minimal path that does not include the current directory (“.”). Including the current directory in root’s path may allow attackers to trick root into running malicious copies of programs. For example, if the current directory is early in root’s path, then an attacker might put a program called `ls` in a writable directory. Should root type `ls` in that directory, the attacker’s program will be run with root privileges. Even if the current directory is late in the path, an attacker might name a program `sl` to capitalize on a common typo.

[1] Sun Microsystems, *SunSolve Home*, <http://sunsolve.sun.com/> (17 March 2003).

[2] Software Engineering Institute, *CERT Coordination Center*, 17 March 2003, <http://www.cert.org> (17 March 2003).

[3] The SANS Institute, “Security Alert Consensus – Archive & Sign-up,” *SANS Institute – Computer Security Education and Information Security Training*, 13 March 2003, <http://www.sans.org/sac> (17 March 2003).

[4] Renaud Deraison <deraison@cvs.nessus.org>, *Nessus*, 28 February 2003, <http://www.nessus.org/> (29 March 2003).

[5] Peter Baer Gavin, “The Solaris Security FAQ,” *Unix Insider*, 1 January, 2001, <http://www.itworld.com/Comp/2377/security-faq> (17 March 2003).

[6] Mark Strong, “Defense In-Depth on a Solaris 2.X System,” *SANS Info Sec Reading Room*, 1 July, 2001, <http://www.sans.org/rr/unix/solaris_2x.php> (17 March 2003).

[7] Hal Pomeranz, *Solaris Security Step By Step Version 2.0* (The SANS Insitute, 2001).

**Finding 3:** **root has a umask of 002. (Moderate)**

**Solution:** **Add the following to /.profile:**

```
umask 007
```

**Finding 4:** **root has an excessive PATH. (Moderate)**

**Solution:** **In `/.profile`, PATH should be set as follows:**

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/bin:/usr/bin
export PATH
```

**`/usr/local` paths are recommended for root because so many source-compiled tools on giacmain have been put into `/usr/local`.**

### 4.4.2 Startup ("rc") files

`/etc/rc2.d` and `/etc/rc3.d` contain scripts that are run at boot time. Only services that are required should be started. If a service is not required, its package should be removed. Sometimes this is not possible as in the case of BIND which is part of the core Solaris package.

For services that cannot be removed, the startup file should be removed or prepended with `.NO` to avoid starting the script and to provide a record that the service was intentionally turned off. Note that there is no magic in the `.NO` prefix. Any script in these directories that does not start with `S` or `K` is ignored during startup. Renaming the script just makes it easier to restore in the future if desired.

**Finding 5:** **There are unnecessary but enabled startup services. (Moderate)**

**Solution:** **Evaluate each entry in `/etc/rc2.d` and `/etc/rc3.d` to determine if it is needed. Unneeded entries should be removed or renamed with a `.NO` prefix.**

### 4.4.3 Protect console devices

Console device permissions are controlled by `/etc/logindevperm`, which should set these permissions to 600. giacmain does this.

### 4.4.4 Turn off routing

Solaris machines provide network routing by default if they have multiple network interfaces. To turn this off, touch `/etc/notrouter`. giacmain has done this.

### 4.4.5 Avoid dynamic routes

Dynamic routing tables are vulnerable to bad routing information, so static routing tables are preferred. giacmain is not running any dynamic routing daemons:

```
# ps -ef | grep routed
root 19117 17737  0 19:51:45 pts/51   0:00 grep routed
# ps -ef | grep in.rdisc
root 18708 17737  0 19:48:19 pts/51   0:00 grep in.rdisc
```

### 4.4.6 Limit write access to `/var/adm/utmpx`

`/var/adm/utmpx` tracks successful and failed logins. Modifying this file can allow an attacker to mask their activities. It should have 644 permissions. This is set correctly on giacmain.

### 4.4.7 Syslog

`/etc/syslog.conf` controls how much information syslog collects. Inspection of this file on giacmain demonstrated that it is logging all message classes reasonably.

### 4.4.8 Stack protection

Many attacks rely on the ability to run things on the program stack and use a technique called "stack smashing" to exploit this ability.[8] Solaris can protect against many simple stack smashing attacks by making the following settings in `/etc/system`:

```
set noexec_user_stack = 1
set noexec_user_stack_log = 1
```

The first option disallows running any code on the stack. The second setting logs attempts to do so. These settings will only protect against simple stack smashers since there are techniques for working around this protection.[9]

Applying these settings may break a few programs that legitimately require executing code from the stack, so careful testing is required before applying this configuration.

[8] Aleph One <aleph1@underground.org> "Smashing the Stack for Fun and Profit," *Phrack* 7, Issue 49 (8 November 1996), <http://www.phrack.org/phrack/49/P49-14> (17 March 2003), File 14.

[9] John McDonald <jmcdonal@unf.edu>, "Defeating Solaris/SPARC Non-Executable Stack Protection," *Bugtraq*, 2 March 1999, <http://www.securityfocus.com/archive/1/12734> (17 March 2003).

**Finding 6: Stack protection is not enabled. (Moderate)**

**Solution: Consider adding the following to `/etc/system`:**

```
set noexec_user_stack = 1
set noexec_user_stack_log
```

**A reboot is required after making this change. Be sure to carefully test important applications, since a few do not work under stack protection.**

### 4.4.9 Login banners

Users should not receive a "welcome" banner when logging in. Instead the banner should explain that only authorized access is permitted and all activity may be logged. For SSH, the login banner is `/etc/motd`. For PureFTPd (which giacmain uses instead of the stock ftpd), the banner is stored in *`<ftproot>`*`/.banner`.

**Finding 7: FTP and login do not have warning banners. (Low)**

**Solution: `/etc/motd` and `/ftp/.banner` should be updated to include warning text.**

### 4.4.10 TCP sequence number generation

Many spoofing attacks are based on guessing the appropriate TCP sequence number[10]. To improve TCP sequence number generation on Solaris, set `TCP_STRONG_ISS` to 2 in `/etc/default/inetinit`.

**Finding 8: `TCP_STRONG_ISS` is set to 1. (Moderate)**

**Solution: `TCP_STRONG_ISS` should be changed to 2 in `/etc/default/inetinit`.**

### 4.4.11 Promiscuous networking

Once an attacker has gained privileged access to a machine, she may reconfigure the network devices to promiscuous mode. In promiscuous mode, the compromised machine acts as a network traffic sniffer and can be used to capture unencrypted network traffic from other machines.

Since there are few good reasons for a network device to be in promiscuous mode, checking for this is a valuable way to detect that a machine has been compromised. Unfortunately, Solaris does not provide any native way to determine whether a device is in promiscuous mode. The program ifstatus[11] provides

[10] Michal Zalewski <lcamtuf@bos.bindview.com>, "Strange Attractors and TCP/IP Sequence Number Analysis," *RAZOR*, 21 April, 2001, <http://razor.bindview.com/publish/papers/tcpseq.html> (17 March 2003).

[11] Rob Thomas <robt@cymru.com>, "Welcome to Rob's Tools and Utilities Page," *Cymru.com*, <http://www.cymru.com/Tools> (17 March 2003).

this functionality. It outputs nothing unless it discovers a promiscuous device, in which case it outputs an error. This makes it appropriate for a cronjob.

**Finding 9: Ifstatus is not running. (Moderate)**

**Solution: Periodically run ifstatus to detect a promiscuous network device.**

### 4.4.12 Open Ports

In order to find unexpected open ports, we ran the Nmap[12] port scanner version 3.00 against giacmain. The full results are detailed in section 7.1, Nmap Scan on page 34. There were no problems found except for the open Veritas Netbackup ports, which are addressed on page 22 (“Finding 23: `/etc/inetd.conf` lists too many services. (Low)”).

| Open Port/ Protocol | Nominal Service | Comments |
|---|---|---|
| 21/tcp | ftp | PureFTPd |
| 22/tcp | ssh | OpenSSH |
| 25/tcp | smtp | Sendmail |
| 53/tcp | domain | BIND |
| 80/tcp | http | Apache |
| 113/tcp | auth | This is a fake version of identd that provides no real information. |
| 443/tcp | https | Apache |
| 587/tcp | submission | Sendmail |
| 5050 | mmcc | Actually RealMedia Server |
| 7070/tcp | realserver | RealMedia Server |
| 8080/tcp | http-proxy | RealMedia Server |
| 9090/tcp | zeus-admin | RealMedia Server |
| 13722/tcp | VeritasNetbackup | Unused backup system |
| 13782/tcp | VeritasNetbackup | Unused backup system |
| 13783/tcp | VeritasNetbackup | Unused backup system |

[12] Fyodor <fyodor@insecure.org>, *Nmap -- Free Stealth Port Scanner For Network Exploration & Security Audits*, 22 February 2003, <http://www.insecure.org/nmap/> (28 March 2003).

To determine what is actually running on these open ports, we use lsof:

```
# lsof -i :7070
COMMAND     PID     USER   FD   TYPE        DEVICE SIZE/OFF NODE NAME
rmserver 16222 mkeohane   11u  IPv4 0x30006e965a8      0t0  TCP giacmain.example.com:7070
(LISTEN)
rmserver 16222 mkeohane   15u  IPv4 0x30006503e78      0t0  TCP localhost:7070 (LISTEN)
```

## *4.5 Access Controls*

### 4.5.1 Controlling root login

Privileged (root) access should be performed by sudo as much as possible. Direct root logins should only be allowed at the physical console for emergency purposes.

To restrict getty root logins to the console, `/etc/default/login` should include:

```
CONSOLE=/dev/console
```

This is set correctly on giacmain.

To restrict root SSH logins, `/etc/ssh/sshd_config` should include:

```
PermitRootLogin no
```

**Finding 10: SSH allows public key root logins. (Moderate).**

**Solution: `/etc/ssh/sshd_config` has `PermitRootLogin` set to `without-password`. This setting allows root logins with public key authentication. This is generally more secure than password authentication, provided that the private key is properly protected, but less secure than setting `PermitRootLogin` to `no`. Consider changing this to `no`.**

### 4.5.2 Passwords

Good passwords are critical to the security of a system. If an attacker can find a username/password combination, they can masquerade as a legitimate user on the system. In the best situation, the attacker has access to the user's data and the system's resources. In the worst situation, the attacker may be able to use a privilege escalation exploit to convert the user account to root access.

There are a three main ways for an attacker to find a username/password combination: sniffing, social engineering and password cracking.

If an attacker has access to the physical network, or can trick a server into routing traffic through a network she does have physical access to, then the attacker can watch all the packets as they go by. This is called "sniffing." In this way, the attacker can gain access to any passwords that are sent unencrypted over the network. Protecting against this is covered in section 4.5.3 "Preventing telnet and rlogin/rsh access."

A social engineer tricks a user into providing the attacker with sensitive information. Social engineering relies on human interaction rather than technical attacks, and so is outside the scope of this document. For a good overview of social engineering and how to protect against it, see Kevin Mitnick's *The Art of Deception*[13].

Password cracking is the practice of finding a user's password by guessing a large number of passwords. To do this effectively on a UNIX system, the attacker needs to gain access to `/etc/shadow` which contains the password hashes. While there is no known way to turn a password hash back into a password, it is possible to guess hundreds of thousands of passwords, hash them, and then compare the hashes. If the hashes match, then the password has been found. There are some other complications, such as password salts, but these are beyond the scope of this document.

Note that `/etc/shadow` is only readable by root, so before an attacker can perform password cracking, she theoretically already would have root access. While this fact does provide significant protection there are still compelling reasons for strong passwords:

- The attacker may only have limited root access, allowing her to read an arbitrary file for instance. A weak password allows the attacker to turn this limited root access into full user access.
- Users often use the same passwords from machine to machine. If the attacker can gain root access on one machine and then crack a given user's password, the attacker then may gain user-level access to other machines.
- Repeated login-failures will cause logging errors and account-lockouts through standard login facilities like telnet and ssh. This is one significant reason that login password guessing is impractical. This is not true of password-protected web pages, particularly pages protected by BasicAuth (the browser-generated pop-up dialog). An attacker can generally perform the hundreds of thousands of password guesses required to break these passwords without generating any warning to the administrators or user. The best defense against this is strong passwords.

John the Ripper[14] is a popular password cracker that can guess tens of thousands of passwords a second and has numerous features to make it easy to attack large password databases. We first ran John the Ripper against giacmain's password database in "single" mode, which assumes that users have used permutations of their username or their full name:

```
# ./unshadow passwd shadow > passwords
# ./john -single passwords
```

[13] Kevin D. Mitnick and William L. Simon, *The Art of Deception* (Wiley Publishing, Indianapolis, 2002).

[14] Openwall Project, "John the Ripper Password Cracker," *OpenWall*, <http://www.openwall.com/john/> (25 March 2003).

After about half an hour, John cracked 177 passwords out of 5662 (3%). We then ran John in dictionary mode using the reasonably small default dictionary:

```
# ./john -wordfile:password.lst -rules passwords
```

This cracked another 184 passwords in approximately one and a half hours, giving us a total of 361 or about 6% of all accounts in about two hours on a Sparc Ultra 5. Using larger dictionaries, such as AccessData's "The Kitchen Sink"[15] and running John for longer on more powerful hardware would clearly crack many more passwords. None of the cracked accounts belonged to administrative staff.

**Finding 11: At least 6% of passwords are trivially crackable. (High)**

**Solution: Implement regular password checking and warn users of bad passwords**

**Consider installing a stricter password-setting utility such as npasswd.**[16]

### 4.5.3 Preventing telnet and rlogin/rsh access

Telnet, rlogin and rsh are extremely insecure methods of login. They pass credentials unencrypted and can be trivially tricked by IP spoofing. None of them should be enabled in /etc/inetd.conf. These services are disabled on giacmain.

As secondary protection, /.rhosts and /etc/hosts.equiv should be empty, and they are on giacmain.

### 4.5.4 Lock unneeded system accounts

Extraneous system accounts should be removed or locked. System accounts should have uids below 100 or above 60,000, a non-usable shell such as /bin/false or /bin/noshell, and should have their home directory on system space rather than user space. The following errors were found:

| Account | UID | Shell | Home Directory |
|---|---|---|---|
| bin | | /bin/csh | /users2/bin |
| nuucp | | /usr/lib/uucp/uucico | |
| smmsp | | /bin/fasle [sic] | |
| msql | 160 | /bin/csh | |
| majordom | 197 | | /users18/majordom |

[15] AccessData, Corp., "The Kitchen Sink," *AccessData*, <http://www.accessdata.com/dictionaries/kitchensink.zip> (25 March 2003).

[16] Clyde Hoover <c.hoover@cc.utexas.edu>, "Npasswd - A better password change program," *ITS Unix Services*, 30 January 2003, <http://www.utexas.edu/cc/unix/software/npasswd/> (25 March 2003).

**Finding 12: Some system accounts are not configured correctly. (Moderate)**

**Solution:** **`bin` should have its shell set to `/bin/false` or noshell, and should not have a user-space home directory.**

**`nuccp` and `smmsp` are not used and should be removed. If not removed, they should have their shell set to `/bin/false` or noshell.**

**`msql` and `majordom` should be renumbered to below 100.**

**`msql` should have its shell set to `/bin/false` or noshell.**

**`majordom` should not have a user-space home directory.**

**Finding 13: Various methods are used to lock accounts. (Low)**

**Solution:** **Accounts are locked with a number of different hash strings, including `NP`, `x`, `*LK*`. One system should be chosen to make searching for locked accounts easier.**

**Finding 14: Non-login accounts omit the shell or use `/bin/false`. (Low)**

**Solution:** **Consider installing and using noshell[17]. This will provide notification of suspicious login attempts.**

### 4.5.5 File permissions

No files in `/etc` or `/usr` should be world writable. We checked this as follows:

```
# find /etc /usr \! -type l -perm +002
#
```

Note that symbolic links always appear to be world writable. This is normal and not a problem, which is why the "`\! -type l`" is required.

### 4.5.6 Setuid/setgid programs

Setuid and setgid programs are common attack vectors. If they can be subverted, the attacker can gain access to the effective account which is often root. Attackers may also create new setuid programs to perform tasks on their behalf or to provide themselves an easy root shell. Therefore, it is important to keep track of setuid and setgid programs, particularly ones in user-writable locations such as user or temporary directories.

[17] Carnegie Mellon University, "Installing noshell to support the detection of access to disabled accounts on systems running Solaris 2.x.," *CERT Coordination Center*, 9 January, 2001, <http://www.cert.org/security-improvement/implementations/i049.02.html> (17 March 2003).

```
# find /tmp /var/tmp /users* -follow -type f \( -perm +2000 -o -perm +4000 \)|xargs ls -l
---sr-x---    1 khayes   www             0 Apr 24  2001 /users14/khayes/WWW/d.txt.dir
---sr-x---    1 khayes   www             0 Apr 11  2001 /users14/khayes/WWW/e.dir
---sr-x---    1 khayes   www             0 Apr 11  2001 /users14/khayes/WWW/e.pag
-rwxr-sr-x    1 pparthas pparthas      153 Jul  9  2002 /users16/pparthas/.Xauthority
-rwxr-sr-x    1 pparthas pparthas      192 Jul 11  2002 /users16/pparthas/.cshrc
-rwxr-sr-x    1 pparthas pparthas      653 Jul 11  2002 /users16/pparthas/.login
-rwxr-sr-x    1 pparthas pparthas      697 Jul 11  2002 /users16/pparthas/.profile
-rwxr-sr-x    1 pparthas pparthas        0 Feb 20  2002 /users16/pparthas/.siteindex
---Sr-x---    1 root     hank         5736 Apr 20  1998 /users18/hank/.elm/become
-rwsr-sr-x    1 hillenda hillenda      569 Dec 19  1997 /users18/hillenda/index.html
-rwsr-xr-x    1 narana   www         96000 Mar  4  1998 /users18/narana/WWW/cgi/Count.cgi
-rw---s---    1 kpanyam  jolee          44 May 10  2002 /users4/kpanyam/lib1/WWW.passwd
-r-s--x--x    1 clark    clark      572452 Sep 13  2000 /users6/clark/kermit
```

Each of these should be checked to make sure it's legitimate. Several are minor mistakes such as the files in /users14/khayes. Some are more serious such as the file in /users16/pparthas which could allow anyone to gain an X session running as pparthas. clark has a similar problem, but that one is not immediately exploitable because /users6/clark is not world readable (as /users16/pparthas is).

The most critical finding is /users18/hank/.elm/become. This is a root setuid script, readable by the hank group. Further investigation shows that this is almost certainly a root-shell exploit:

```
# file become
become: ELF 32-bit MSB executable SPARC Version 1, dynamically linked, not stripped
# strings become
uid = %d
/usr/local/bin/tcsh
```

**Finding 15: There is a root-shell exploit. (High)**

**Solution: /users18/hank/.elm/become is a root-shell exploit. This must be removed immediately and the matter investigated. This exploit has existed for 5 years and may or may not be actively used.**

**Finding 16: There are mistaken setuid files in user home directories. (High)**

**Solution: These should be fixed immediately and the users notified to prevent this happening in the future. These files are owned by users so the impact is less severe and are almost certainly user error.**

When developing new setuid programs, they must be very carefully written to avoid security flaws. Whenever writing a setuid/setgid program, consider the following questions:

1. Could this functionality be provided by group permissions? For example, if a user needs to update a particular file, could the file be group-owned by a group that the user belongs to?

2. Could this functionality be provided by a setgid program instead of a setuid program? If limited file access is needed, setgid programs are generally sufficient.
3. Could this functionality be provided by a setuid program owned by a user other than root? It may be worth creating a dummy user specifically for this functionality.
4. Could this functionality be provided by a proxy user? Rather than making a program setuid-root, consider making a dummy user, granting that user specific root sudo privileges which the program uses, and then making the program setuid to that user.

## *4.6 Administrative practices*

### 4.6.1 Change management

A comprehensive change management system helps improve security by providing a clear history of security decisions. Knowing when and why changes were made to the system is invaluable in ensuring that security choices aren't undermined by later patches or uncoordinated changes.

giacmain maintains some configuration files with RCS[18], but has no comprehensive change management strategy. Since only some files are managed with RCS, administrators often forget to check out files before changing them, leading to untracked changes.

One simple solution is to ensure that every configuration file is under RCS. This will improve tracking and will make administrators more likely to remember to check out files, since every file they edit will need to be checked out.

A more complex solution is to install a configuration system like cfengine[19]. This will help centralize all configuration decisions, better ensuring that they are maintained through patch upgrades or "temporary" changes (which are generally more permanent than intended). Maintaining the cfengine configuration files on another system will also provide a simple form of system monitoring, helping detect configuration-modifying attacks.

[18] Free Software Foundation, "RCS," *GNU's Not Unix!*, 8 February 2003, <http://www.gnu.org/software/rcs/rcs.html> (17 March 2003).

[19] Mark Burgess <mark@iu.hio.no>, *Cfengine – a configuration engine for Unix and Windows*, <http://www.cfengine.org> (17 March 2003).

**Finding 17: There is no consistent change management system. (Low)**

**Solution: Manage all configuration files under RCS or CVS, or install a more comprehensive system such as cfengine.**

### 4.6.2 Monitoring

Systems should be actively monitored in order to detect and respond quickly to attacks. There are two primary types of monitoring: change monitoring and log monitoring.

#### Change monitoring

Change monitoring alerts administrators to changes in system configuration. This helps detect successful attacks, but also provides an audit record of changes to the system. In the absence of a comprehensive change management system (see Change management, p. 17), this can help administrators determine at least when certain changes were made. The most popular change monitoring tool is Tripwire[20]. Tripwire keeps checksums of all system files on protected media, ideally hardware-enforced non-writable media such as a CD. Tripwire then periodically compares the current system to the archived checksums. If changes are found, the administrators are notified. This will detect many types of attacks, including user-space root kits. It will not detect kernel-level root kits unless the attacker is sloppy.

**Finding 18: There is no change monitoring system. (Moderate)**

**Solution: Consider deploying Tripwire. Note that the Open Source version of Tripwire has been ported to Solaris 8 by Paul Herman <pherman@frenchfries.net> on his Tripwire™ Patches[21] page.**

#### Log monitoring

Log monitoring alerts administrators to significant or unusual system activity. Ideally this can allow the administrators to foil an attack before it is successful.

Giacmain has a custom-written log monitoring system that relies on syslog filtering important information into `/var/adm/problems`. It sends out log excerpts once a day. No facilities exist for detecting attacks in-progress.

A more comprehensive solution is a log monitoring package such as Logwatch[22]. Logwatch can be configured to scan any log and send periodic summary reports.

[20] Tripwire Open Source Project, T*ripwire.org Home Page*, <http://www.tripwire.org> (17 March 2003).

[21] Paul Herman <pherman@frenchfries.net>, "Tripwire™ Patches," *Paul Herman's Home Page*, 5 November 2001, <http://www.frenchfries.net/paul/tripwire/> (18 March 2003).

[22] Kirk Bauer <kirk@kaybee.org>, *Logwatch*, 28 March 2002, <http://www.logwatch.org/> (18 March 2003).

**Finding 19: There is no comprehensive log monitoring. (Moderate)**

**Solution: Giacmain monitors its logs with homegrown scripts relying on all interesting messages going to `/var/adm/problems`. Consider deploying a more comprehensive log monitoring tool such as Logwatch.**

### 4.6.3 Privileged Access

Privileged (root) access should be limited as much as possible. There are several ways that individuals gain root access: root password, sudo and setuid/setgid programs.

#### Root Password

The root password should be carefully protected and used only when absolutely necessary, such as when booting into single user mode. Not all administrators may need access to the root password. Only two of the eight giacmain administrators have access to the root password, and its use is avoided.

#### Sudo

Sudo[23] provides root and other privileged access to particular users for particular commands. Users authenticate with their own password, so they do not need access to the root password. Besides root, users can be granted access to other effective user ids such as the web server, allowing finer control. User access can also be limited to particular commands. All commands run via sudo are logged, providing an audit trail.

Per-command access with sudo should be carefully considered. There are many ways to escalate limited sudo access into full root access. Apparently innocent commands such as chmod can easily provide a root shell. Any command that allows shell-outs, such as vi, can also provide arbitrary root access. In general, users granted sudo access must be trusted not to abuse it, or the needed functionality should be wrapped into a carefully written script to protect from abuse.

Sudo is configured with a flat file, `/usr/local/etc/sudoers`, which should be readable only by root, as it is on giacmain:

```
# ls -l /usr/local/etc/sudoers
-r--r-----    1 root     root         1646 Dec  2 12:06 /usr/local/etc/sudoers
```

Giacmain makes extensive use of sudo for administrative activities and privilege separation for setuid programs. Some users have also been granted limited sudo privileges in order to maintain particular packages. In general this has been done reasonably. The only major issue is that administrative root access is granted without a password:

```
ADMINS ALL=(ALL) NOPASSWD: ALL
```

[23] Todd Miller <Todd.Miller@courtesan.com>, *Sudo Main Page*, 25 April 2002, <http://www.courtesan.com/sudo/> (17 March 2003).

While this is convenient for administrators, it is extremely dangerous. Any attacker who can gain access to any of the administrators' accounts can have unlimited root access. The attacker does not have to gain access to the administrator's password. She might find an administrator's logged-in terminal, trick the administrator into running a command or break an administrator's private SSH key. Administrators should be required to authenticate with a password before running root commands. An even better solution is to give each administrator a separate account for root access; for example `frank` would also have `frank-adm` for root access. This prevents commingling of the roles of user and administrator and allows the administrative account to be better protected.

**Finding 20: Administrative sudo access does not require a password. (High)**

**Solution: Remove the `NOPASSWD` option in `/usr/local/etc/sudoers`.**

**Finding 21: Administrative access is performed with user accounts. (Low)**

**Solution: Administrators log in with their own accounts and then use sudo. Consider providing administrators separate administrative accounts to help them better separate their user and administrative functions.**

### Setuid/setgid Programs

For information on setuid/setgid programs, see Section 4.5.6 "Setuid/setgid programs" on page 15.

### 4.6.4 Backups and disaster recovery

All critical systems should have regular, tested backups and a tested disaster recovery plan. Even a rudimentary backup system will dramatically improve the ability to recover from common mishaps.

A popular though complex backup solution is Amanda[24]. Amanda is quite good at managing moderate-sized networks with a centralized backup server. For a small number of machines, however, setting up Amanda can be daunting.

For single machines with their own backup devices, afio[25] makes it easy to create a simple backup solution. Probably the simplest wrapper for afio is dobackup.pl,[26] which aims to handle everything but changing the tapes.

Since giacmain is critical to nearly every employee at GIAC Enterprises, its disaster recovery plan should be carefully integrated into GIAC Enterprises'

[24] University of Maryland at College Park, *Amanda, The Advanced Maryland Automatic Network Disk Archiver*, 25 January 2003, <http://www.amanda.org> (24 March 2003).

[25] Koen Holtman <koen@hep.caltech.edu>, "afio," *Freshmeat*, 4 December 2002, <http://freshmeat.net/projects/afio/> (24 March 2003).

[26] Robert Hardy and Ian E. Morgan, "DOBACKUP.PL," *Webcon, Inc.*, 2 February 2002, <http://www.webcon.ca/opensource/dobackup/> (24 March 24, 2003).

overall disaster recovery plan. Currently there is no such plan. The development of one is beyond the scope of this document since it includes numerous non-technical issues such as vendor notification, physical security considerations, public relations and other issues.

| | |
|---|---|
| **Finding 22:** | **There are no regular backups or disaster recovery plan. (High)** |
| **Solution:** | **Immediately establish a regular backup procedure, regular restore testing, and a disaster recovery plan. Webcon's `dobackup.pl` script and University of Maryland at College Park's Amanda are good choices to consider.** |

# 5 Application Detailed Analysis

## 5.1 NTP

An accurate system clock is important to security for several reasons. As discussed in Rick Farrow's "Beating the Clock on Security,"[27] if an attacker can push the system clock back, certain random-number generators may replay a sequence, or expired Kerberos tickets or digital signatures may be accepted. Disrupting the clock can also prevent event correlation engines from detecting simultaneous events. To protect against these kinds of attacks, systems should use the Network Time Protocol (NTP). Giacmain is running a properly configured version of NTP. For more information on NTP, see the Network Time Protocol Project.[28]

## 5.2 Cron

Visual inspection of cronjobs in `/var/spool/cron/crontabs` for adm, root, and sys discovered no issues. There are no at jobs scheduled by privileged users:

```
# ls -l /var/spool/cron/atjobs/
total 3
r-Sr—r--    1 tipetty  tipetty      2358 Mar  8 10:02 1047232946.a
```

## 5.3 Inetd

Inetd provides on-demand services, which means that processes are spawned whenever they are requested from a network socket. Most of the services provided in the stock `/etc/inetd.conf` aren't needed, and this file should be trimmed as much as possible. Removing entries from this list rather than commenting them out makes auditing this list easier in the future.

**Finding 23: `/etc/inetd.conf` lists too many services. (Low)**

**Solution: `/etc/inetd.conf` has entries for `ftp`, `bpcd`, `vopied`, and `bpjava-msvc`. Of these, only `ftp` is used. `bpcd`, `vopied` and `bpjava-msvc` should be removed from `/etc/inetd.conf`.**

Xinetd[29] is an inetd replacement that provides several advantages including improved denial of service protection, improved logging and service forwarding. Its configuration is very different from inetd, and given giacmain's limited use of inetd careful consideration should be given before converting. In general, how-

[27] Rik Farrow <rik@spirit.com>, "Beating the Clock on Security," *Network Magazine* September 2001, <http://www.networkmagazine.com/article/NMG20010823S0006> (17 March 2003).

[28] The NTP Web-Dudes <webdudes@ntp.org>, *The Network Time Protocol Project*, 14 February 2003, <http://www.ntp.org> (5 April 2003).

[29] Rob Braun <bbraun@synack.net>, *xinetd*, <http://www.xinetd.org/> (17 March 2003).

ever, the conversion is easy and the advantages outweigh the cost. A good overview of xinetd is available in curator's Unofficial Xinetd Tutorial.[30]

**Finding 24: Giacmain is using inetd. (Low)**

**Solution: Consider changing to xinetd.**

## *5.4 Sendmail*

Giacmain is running a custom-compiled version of sendmail 8.12.8, which was the latest version at the time of the audit. This version fixes a significant security issue in earlier versions.

Very old versions of sendmail included a `decode` entry in `/etc/mail/aliases`. This entry could be used to write to arbitrary files in the system. While this entry has not been in the default configuration for a long time, older systems may still have this entry if they maintained their old aliases file.[31] Giacmain does not have this problem:

```
$ grep decode /etc/mail/aliases
$
```

Since sendmail can be forced to write arbitrarily large files to `/var/mail` there is a danger of attackers filling this disk. If `/var/mail` and other logging directories such as `/var/adm` and `/var/log` share the same partition, an attacker could cause the system to stop logging. Giacmain separates `/var/mail` onto its own partition:

```
$ df /var/mail
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/dsk/c2t11d0s0    4178456   3741323    395349  91% /var/mail
```

Sendmail should not be setuid root, but should be setgid smmsp to allow it to write to the mail queue directory. In order to bind to port 25 among other things, the sendmail daemon does need to be started as root. Furthermore, the mail queue and configuration files should be protected. This is correctly set on giacmain:[32]

```
$ ls -ld /usr/lib/sendmail /var/spool/clientmqueue /var/spool/mqueue
/etc/mail/sendmail.cf /etc/mail/submit.cf
-r--r--r--    2 root     root        57205 Mar  3 13:19 /etc/mail/sendmail.cf
-rw-r--r--    1 root     other       38953 Jul  2  2002 /etc/mail/submit.cf
-r-xr-sr-x    2 root     smmsp      993900 Mar  3 10:29 /usr/lib/sendmail*
drwxrwx---    2 smmsp    smmsp      580608 Mar 28 16:46 /var/spool/clientmqueue/
drwx------    3 root     bin        142848 Mar 28 16:46 /var/spool/mqueue/
```

[30] curator <curator@macsecurity.org>, "An Unofficial Xinetd Tutorial," *Xinetd for OSX*, <http://www.macsecurity.org/resources/xinetd/tutorial.shtml> (17 March 2003).

[31] SAINT Corporation, "Sendmail Decode Vulnerability," *Tutorial - Sendmail Information*, <http://www.saintcorporation.com/demo/saint_tutorials/sendmail_decode.html> (17 March 2003).

[32] Sendmail, Inc., *sendmail/SECURITY*, 23 September 2002, <http://www.sendmail.org/secure-install.html> (28 March 28, 2003), "sendmail configuration without set-user-ID root."

## 5.5 FTP

Giacmain has installed a replacement FTP package called PureFTPd[33]. However, the Sun FTP packages have not been removed.

**Finding 25: `SUNWftpr` and `SUNWftpu` are installed but not used. (Low)**

**Solution: `SUNWftpr` and `SUNWftpu` should be removed to avoid accidentally turning on the stock ftpd or making unexpected changes due to patches in these packages.**

PureFTPd is in more secure "out-of-the-box" than most other FTP servers such as Solaris' default FTP server or WU-FTPD.[34] Even so, care should be exercised when setting up PureFTPd. The most significant non-default security options for PureFTPd are:

- `--chrooteveryone`: This forces everyone but root into a chrooted environment, which is a good precaution for a fundamentally insecure protocol such as FTP. Giacmain has this set.
- `--anonymouscantupload`: This prevents attackers from taking over your users' FTP sites to post software, music, pornography, attack tools or other content. Even if users make their directories world-writable, PureFTPd will not permit writing by anonymous users. Giacmain has this set.
- `--prohibitdotfilesread`: This prevents the reading (and writing) of "dot files" by users, which can help prevent attackers who have hijacked an FTP session from escalating that into full account access. Conversely, it can be frustrating to users who are trying to manage web sites with FTP, since it prevents writing `.htaccess` files. This may still be considered a benefit since it helps encourage use of SSH-based protocols instead of insecure FTP solutions. Giacmain has this set.
- `--with-privsep`: The latest version of PureFTPd, 1.0.14, adds privilege separation which improves overall security of the system by limiting the amount of code that has root access.

**Finding 26: Giacmain is running an old version of PureFTPd. (Low)**

**Solution: Upgrade to PureFTPd 1.0.14 to gain improved privilege separation facilities.**

## 5.6 Apache

The Apache web server can be one of the most challenging systems to secure. Since it is both ubiquitous and complex, it is an obvious point of attack. Giacmain

[33] Frank Denis <j@pureftpd.org>, *PureFTPd – A fast, standard compliant, production quality FTP server*, 30 January 2003, <http://www.pureftpd.org/> (24 March 2003).

[34] WU-FTPD Development Group, *WU-FTPD Development Group*, 18 February 2003, <http://www.wu-ftpd.org/> (24 March 2003).

is running Apache version 2.0.43. The most recent version of Apache at the time of the audit was 2.0.44. None of the security fixes between 2.0.43 and 2.0.44 apply to Solaris, and none of the bug fixes are significant for giacmain, so 2.0.43 is an acceptable release.

### 5.6.1 Permissions

Apache should run as an unprivileged (non-root) user to limit what can be done if an attacker compromises the daemon. This account should not have write-access to the Apache configuration files; otherwise an attacker who has compromised the daemon could reconfigure the system, potentially causing the daemon to run as root in the future. This is true on giacmain:

```
# find /var/apache -user www
# find /var/apache –group www
# find /var/apache –perm +o=w
#
```

Care must be taken with the programs that Apache uses, such as `apachectl`, `httpd`, `logrotate`, and `apachectl`. On giacmain, these are stored in `/usr/local/packages/apache/apache-2.0.43/bin` and are linked into `/usr/local/bin`. These directories, as well as all parent directories, should only be writable by root. This is true on giacmain.

```
$ cd /usr/local/packages/apache/apache-2.0.43/bin
$ ls -l
total 8272
rwxr-xr-x    1 root     bin        147560 Dec  4 19:56 ab*
rwxr-xr-x    1 root     bin          5002 Dec  4 19:53 apachectl*
rwxr-xr-x    1 root     bin         20588 Dec  4 19:53 apxs*
rwxr-xr-x    1 root     bin         56148 Dec  4 19:56 checkgid*
rwxr-xr-x    1 root     bin         10683 Dec  4 19:53 dbmmanage*
rw-r--r--    1 root     bin           215 Jul  9  2002 envvars
rw-r--r--    1 root     bin           215 Dec  4 19:53 envvars-std
rwxr-xr-x    1 root     bin        103288 Dec  4 19:56 htdbm*
rwxr-xr-x    1 root     bin         64460 Dec  4 19:56 htdigest*
rwxr-xr-x    1 root     bin         99020 Dec  4 19:56 htpasswd*
rwxr-xr-x    1 root     bin       7728604 Dec  4 19:56 httpd*
rwxr-xr-x    1 root     bin         60340 Dec  4 19:56 logresolve*
rwxr-xr-x    1 root     bin         62980 Dec  4 19:56 rotatelogs*
rwsr-xr-x    1 root     bin         51128 Dec  4 19:56 suexec*
$ ls -ld /usr /usr/local /usr/local/packages /usr/local/packages/apache
/usr/local/packages/apache/apache-2.0.43 /usr/local/packages/apache/apache-2.0.43/sbin
drwxr-xr-x   31 root     sys          1024 Jul 16  2002 /usr/
drwxr-xr-x   18 root     root          512 Mar  1 14:32 /usr/local/
drwxr-sr-x  103 root     bin          2048 Feb 26 06:00 /usr/local/packages/
drwxr-sr-x    4 root     bin           512 Jan 15 12:26 /usr/local/packages/apache/
drwxr-sr-x    8 root     bin           512 Jul 29  2002
/usr/local/packages/apache/apache-2.0.43/
drwxr-sr-x    2 root     bin           512 Dec  4 19:56
/usr/local/packages/apache/apache-2.0.43/bin/
$ ls -ld /usr/local/sbin
drwxr-xr-x    2 root     other        1024 Jan 29 12:22 /usr/local/sbin/
```

### 5.6.2 User/Group

As mentioned earlier, Apache should run as an unprivileged user and group. This is true on giacmain:

```
$ egrep ‘^User |^Group ‘ /var/apache/conf/httpd.conf
User www
Group www
$ grep ^www: /etc/passwd /etc/group
/etc/passwd:www:x:39:39:Apache Admin:/usr/local/packages/apache:/bin/false
/etc/group:www::39:
```

### 5.6.3 Options

Many important options are controlled by the `Options` directive. Care should be taken regarding the scope of any particular `Options` directive, since these are `Location` and `Directory` specific.[35]

#### Symbolic Link Handling

Symbolic link handling is extremely important for a secure system. If the web server follows user-created symbolic links, the user may be able to gain access to any file the web server has access to, including files that would normally be protected by `AuthBasic` directives.

Disallowing symbolic links is the most secure configuration but quite limiting. A trade-off is the `SymLinksIfOwnerMatch` option. With this set, a symbolic link is only followed if the owner of the symbolic link is the same as the owner of the target. This ensures that the user can only have the web server access files the user already has access to. `SymLinksIfOwnerMatch` is extremely expensive in terms of performance, even if symbolic links are not used, so should generally only be used in user-writable areas. Giacmain does this. Note that even with an `AllowOverride`, symbolic link handling cannot be overridden.

```
<Directory /users*/*/WWW>
AllowOverride FileInfo AuthConfig Limit Indexes Options
Options MultiViews -Indexes SymLinksIfOwnerMatch IncludesNoExec ExecCGI
[…]
```

#### Server Side Includes

Server side includes[36] (SSI) are a powerful way of providing common elements throughout a website such as consistent headers and footers. SSIs are also a potential security threat due to the `#exec` element which will cause Apache to execute an arbitrary program as the web server. Since giacmain allows SSIs in user-written web pages, the `#exec` element should not be allowed. This is controlled by the `IncludesNOEXEC` option, and should be set for all user-writable directories. Giacmain does this. Even though the user is allowed to override `Options` due to the `AllowOverride`, `IncludesNoExec` overrides any `+Includes` that the user might put in an `.htaccess` files.

[35] Apache HTTP Server Documentation Project, “Configuration Sections,” *Apache HTTP Server Version 2.0 Documentation*, <http://httpd.apache.org/docs-2.0/sections.html> (17 March 2003).

[36] Apache, “Apache Tutorial: Introduction to Server Side Includes,” *Documentation*, <http://httpd.apache.org/docs-2.0/howto/ssi.html> (17 March 2003).

```
<Directory /users*/*/WWW>
AllowOverride FileInfo AuthConfig Limit Indexes Options
Options MultiViews -Indexes SymLinksIfOwnerMatch IncludesNoExec ExecCGI
[…]
```

### Indexes

By default, requests for a directory where there is no index file (`index.html`, `index.htm`, etc.) return an automatically generated file list, including file types and other file system information. This is extremely useful in some cases but should be turned off by default to prevent attackers from discovering more information about the file system than they should. This functionality is controlled by the `Indexes` option. Giacmain turns this off globally but allows users to turn it back on for particular directories, which is reasonable trade-off.

```
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
[…]
<Directory "/var/apache/htdocs">
Options -Indexes FollowSymLinks ExecCGI
[…]
<Directory /users*/*/WWW>
AllowOverride FileInfo AuthConfig Limit Indexes Options
Options MultiViews -Indexes SymLinksIfOwnerMatch IncludesNoExec ExecCGI
[…]
```

The `Options` directive for the root directory (`/`) is absolute which disables indexing globally.[37] The main `DocumentRoot` (`/var/apache/htdocs`) explicitly turns indexing off again. User directories also explicitly turn off indexing, but allow users to turn it back on with `.htaccess` files if they need it.

## 5.6.4 Access Control Lists

By default, the web server will serve any file that it can access. To change this behavior requires Allow and Deny directives.[38] The root directory should by default deny all access in order to protect the main file system. Individual directories should then allow access. This is accomplished with the following:[39]

```
<Directory />
Order Deny,Allow
Deny from all
</Directory>
```

[37] Apache, "Apache Core Features," *Documentation,* <http://httpd.apache.org/docs-2.0/mod/core.html#options> (17 March 2003), Options.

[38] Apache, "Apache Module mod_access," *Documentation,* <http://httpd.apache.org/docs-2.0/mod/mod_access.html> (17 March 2003).

[39] Apache, "Security Tips," *Documentation,* http://httpd.apache.org/docs-2.0/misc/security_tips.html#protectserverfiles (17 March 2003), "Protect Server Files by Default."

**Finding 27: Apache does not have default-deny on the root file system. (Low)**

**Solution: Add the following to `httpd.conf`:**

```
<Directory />
Order Deny, Allow
Deny from all
</Directory>
```

### 5.6.5 User-controllable Configuration

Users can override system configuration using `.htaccess` files. When possible, these should be disabled, but giacmain requires them due to its many independent users.

The `AllowOverride` directive controls what can be placed in `.htaccess` files. Giacmain limits this to `None` at the root directory, and then allows overrides in the user directories:

```
<Directory /users*/*/WWW>
AllowOverride FileInfo AuthConfig Limit Indexes Options
[…]
```

Given the authorization functions of `.htaccess` files, they should generally be protected from browsing. Giacmain does this:

```
<Files ~ "^\.ht">
Order allow,deny
Deny from all
</Files>
```

### 5.6.6 Obfuscation

Obfuscation of the server's configuration is a controversial protection scheme. The intent of obfuscation is to prevent attackers from determining details about the web server configuration including the web server software and version. Lying about this information may deter some types of automated attacks, but most attacks today are either too smart or too dumb to be deterred this way. As described in Detecting and Defending against Web-Server Fingerprinting,[40] smart attacks determine server information using behavior inherent to the server such as HTTP header order and how errors are handled, which is extremely difficult to obscure without introducing a maintainability headache. Dumb attacks attack everything whether it is vulnerable or not. Even so, there may still be a small advantage to obscuring the version information.

There is no way to change Apache's header identification information through simple configuration. This information must be changed in the source code before compiling.

[40] Dustin Lee, Jeff Rowe, Calvin Ko, Karl Levitt, "Detecting and Defending against Web-Server Fingerprinting," *Annual Computer Security Applications Conference*, 12 December 2002, <http://www.acsac.org/2002/papers/96.pdf> (17 March 2003).

1. In `os/unix/os.h`, change `PLATFORM`.
2. In `include/ap_release.h`, change `AP_SERVER_BASEVENDOR`, `AP_SERVER_BASEPRODUCT`, `AP_SERVER_MAJORVERSION`, `AP_SERVER_MINORVERSION`, and `AP_SERVER_PATCHLEVEL`.

The error documents also uniquely identify each vendor. To hide these, create new pages for each 4xx and 5xx error code as defined in the `status_lines` array in `modules/http/http_protocol.c` and use the `ErrorDocument` directive to override the internal error messages:

```
ErrorDocument 400 errors/400.html
ErrorDocument 401 errors/401.html
[…]
```

For explanations of these codes, see RFC2616,[41] RFC2295,[42] RFC2518,[43] and RFC2774.[44]

Once again, these steps will only provide marginally increased security and may make it more difficult to find out-of-date servers on your own network. Doing this tends to make people feel better though.

## *5.7 BIND*

The Berkeley Internet Name Domain server, better known as BIND or `named`, is a perennial source of security vulnerabilities. This is extremely disturbing considering it is basically a name-to-number lookup table that wouldn't even need root access if it didn't bind to a low-numbered port. There are fundamental security vulnerabilities in the DNS protocol, but in this paper we will only address things that administrators can do to protect their systems from basic DNS attacks and root exploits through `named`. An excellent template for setting up a BIND system is Rob Thomas's "Secure BIND Template".[45]

The most important step in protecting the rest of the system from BIND is to chroot it and remove its root privileges by using the `-u` parameter as described in the "Secure BIND Template." Giacmain has done this.

Once BIND is "jailed," attention should be turned to the configuration file. First, zone transfers and updates should be prohibited unless required with the following directives in each zone:

[41] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999, <http://rfc.net/rfc2616.html#s10.4> (17 March 2003), 65-71.

[42] K. Holtman, A. Mutz, *Transparent Content Negotiation in HTTP*, March 1998, <http://rfc.net/rfc2295.html#s8.1> (17 March 2003), 25.

[43] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, *HTTP Extensions for Distributed Authoring – WEBDAV*, February 1999, <http://rfc.net/rfc2518.html#s10.3> (17 March 2003), 60.

[44] H. Nielsen, P. Leach, S. Lawrence, *An HTTP Extension Framework*, February 2000, <http://rfc.net/rfc2774.html#p8> (17 March 2003), 8.

[45] Thomas, "Secure BIND Template Version 3.7," *Cymru.com*, 13 Feburary 2003, <http://www.cymru.com/Documents/secure-bind-template.html> (24 March 2003).

```
allow-transfer { none; };
allow-update   { none; };
```

Unless you use RFC1918[46] address, you should block them. Attackers are likely to spoof these addresses, and there's no legitimate way to receive these addresses from the internet. See the `bogon` access control list in the "Secure BIND Template" for a list of RFC1918 addresses.

**Finding 28: BIND does not block RFC1918 addresses. (Low)**

**Solution: "Blackhole" `bogon` access control list from "Secure BIND Template" in `/var/named/etc/named.conf`.**

Recursive queries request the server do all the work required to resolve a name rather than just passing back the next step in the search. This allows the server to better manage a cache for its clients and improves performance overall. Servers should not provide recursive queries to the internet at large for both performance and security reasons. If a requester can get a DNS server to perform a recursive query for a third-party host (i.e. a host that is not controlled by the targeted name server), the attacker has a reasonable chance of being able to perform a cache poisoning attack.[47] To help protect against this, DNS servers should only perform recursive queries for their own clients.

**Finding 29: BIND accepts recursive queries from unknown hosts. (High)**

**Solution: Add `allow recursion` directive to the `options` section of `/var/named/etc/named.conf` to limit recursive queries to known hosts.**

## *5.8 SSH*

SSH is a critical replacement for telnet and r-services (rsh, rexec, rlogin, etc). Giacmain has shut off all of these insecure services and replaced them with OpenSSH.[48] For the most part, SSH is secure "out-of-the-box," but there are a few things that can improve its security.

A recent addition to OpenSSH is privilege separation. This separates the SSH daemon into two process, a very small privileged process and a larger unprivileged process. Separating privileges this way helps prevent exploits based on errors in the larger unprivileged process. By keeping the privileged process small and simple, hopefully flaws can be minimized. Giacmain is using process separation.

It is also critical that the configuration files are not world-writable, and that the private key is only readable by root. Giacmain does this:

[46] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, *Address Allocation for Private Internets*, February 1996, <http://rfc.net/rfc1918.html> (24 March 2003).

[47] Joe Stewart <jstewart@lurhq.com>, "DNS Cache Poisoning – The Next Generation," *SecurityFocus*, 27 January 2003, <http://www.securityfocus.com/guest/17905> (29 March 2003).

[48] OpenBSD, *OpenSSH*, 4 March 2003, <http://www.openssh.com/> (24 March 2003).

```
$ ls -la /etc/ssh/
total 108
drwxr-xr-x    2 root     other         512 Aug  1  2002 ./
drwxr-xr-x   45 root     sys          4096 Mar 24 18:02 ../
-rw-r--r--    1 root     sys         88039 Jun 27  2002 moduli
-rw-r--r--    1 root     sys          1199 Jun 27  2002 ssh_config
-rw-------    1 root     other         668 Jul  1  2002 ssh_host_dsa_key
-rw-r--r--    1 root     other         608 Jul  1  2002 ssh_host_dsa_key.pub
-rw-------    1 root     other         533 Jul  1  2002 ssh_host_key
-rw-r--r--    1 root     other         337 Jul  1  2002 ssh_host_key.pub
-rw-------    1 root     other         887 Jul  1  2002 ssh_host_rsa_key
-rw-r--r--    1 root     other         228 Jul  1  2002 ssh_host_rsa_key.pub
-rw-r--r--    1 root     sys          2496 Aug  1  2002 sshd_config
```

Note that the directory, `ssh_config` and `sshd_config` are only writable by root, and that all of the private keys are only readable by root.

# 6 Critical Issues and Recommendations

## *6.1 There is a root-shell exploit.*

<table>
<tr><th>Finding 15 (Page 16)</th><th>Severity: High</th></tr>
<tr><td colspan="2"><code>/users18/hank/.elm/become</code> is a root-shell exploit. This must be removed immediately and the matter investigated. This exploit has existed for 5 years and may or may not be actively used.<br><em>Note: This has already been addressed by the administrators.</em></td></tr>
</table>

## *6.2 At least 6% of passwords are trivially crackable.*

<table>
<tr><th>Finding 11 (Page 14)</th><th>Severity: High</th></tr>
<tr><td colspan="2">Implement regular password checking on giacmain and warn users of bad passwords.<br>Consider installing a stricter password-setting utility such as npasswd .</td></tr>
</table>

## *6.3 There is no comprehensive strategy to patching.*

<table>
<tr><th>Finding 2 (Page 7)</th><th>Severity: High</th></tr>
<tr><td colspan="2">Recommended and Security patches should be applied regularly from SunSolve. Due to the large number of source-built packages on giacmain, the administrative staff should also subscribe to the security lists from Sun, CERT and the SANS alert consensus newsletter. Furthermore, they should maintain a list of all source-built packages to compare against announcements from these lists.</td></tr>
</table>

## *6.4 There are no regular backups or disaster recovery plan.*

<table>
<tr><th>Finding 22 (Page 21)</th><th>Severity: High</th></tr>
<tr><td colspan="2">Immediately establish a regular backup procedure, regular restore testing, and a disaster recovery plan. Webcon’s <code>dobackup.pl</code> script and University of Maryland at College Park’s Amanda are good choices to consider</td></tr>
</table>

## *6.5 There are mistaken setuid files in user home directories.*

<table>
<tr><th>Finding 16 (Page 16)</th><th>Severity: High</th></tr>
<tr><td colspan="2">These should be fixed immediately and the users notified to prevent this happening in the future. These files are owned by users so the impact is less severe and are almost certainly user error.</td></tr>
</table>

## 6.6 BIND accepts recursive queries from unknown hosts.

| Finding 29 (Page 30) | Severity: High |
|---|---|
| Add allow recursion directive to the options section of /var/named/etc/named.conf to limit recursive queries to known hosts. | |

## 6.7 Administrative sudo access does not require a password.

| Finding 20 (Page 20) | Severity: High |
|---|---|
| Remove the NOPASSWD option in /usr/local/etc/sudoers. | |

## 6.8 There is no change monitoring system.

| Finding 18 (Page 18) | Severity: Moderate |
|---|---|
| Consider deploying Tripwire. Note that the Open Source version of Tripwire has been ported to Solaris 8 by Paul Herman <pherman@frenchfries.net> on his Tripwire™ Patches page. | |

## 6.9 Some system accounts are not configured correctly.

| Finding 12 (Page 15) | Severity: Moderate |
|---|---|
| bin should have its shell set to /bin/false or noshell, and should not have a user-space home directory.<br>nuccp and smmsp are not used and should be removed. If not removed, they should have their shell set to /bin/false or noshell.<br>msql and majordom should be renumbered to below 100.<br>msql should have its shell set to /bin/false or noshell.<br>majordom should not have a user-space home directory. | |

## 6.10 TCP_STRONG_ISS is set to 1.

| Finding 8 (Page 10) | Severity: Moderate |
|---|---|
| TCP_STRONG_ISS should be changed to 2 in /etc/default/inetinit. | |

# 7 Appendices

## 7.1 Nmap Scan

```
$ sudo nmap -P0 -v -O giacmain.example.com

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
No tcp,udp, or ICMP scantype specified, assuming SYN Stealth scan. Use -sP if you really
don't want to portscan (and just want to see what hosts are up).
Host giacmain.example.com (10.0.0.167) appears to be up ... good.
Initiating SYN Stealth Scan against giacmain.example.com (10.0.0.167)
Adding open port 21/tcp
Adding open port 13783/tcp
Adding open port 110/tcp
Adding open port 8080/tcp
Adding open port 113/tcp
Adding open port 53/tcp
Adding open port 80/tcp
Adding open port 3306/tcp
Adding open port 5050/tcp
Adding open port 995/tcp
Adding open port 7070/tcp
Adding open port 22/tcp
Adding open port 13722/tcp
Adding open port 443/tcp
Adding open port 999/tcp
Adding open port 25/tcp
Adding open port 13782/tcp
Adding open port 5901/tcp
Adding open port 587/tcp
Adding open port 9090/tcp
The SYN Stealth Scan took 60 seconds to scan 1601 ports.
For OSScan assuming that port 21 is open and port 1 is closed and neither are firewalled
Interesting ports on giacmain.example.com (10.0.0.167):
(The 1520 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
25/tcp     open        smtp
53/tcp     open        domain
80/tcp     open        http
113/tcp    open        auth
443/tcp    open        https
445/tcp    filtered    microsoft-ds
587/tcp    open        submission
1433/tcp   filtered    ms-sql-s
2000/tcp   filtered    callbook
2001/tcp   filtered    dc
2002/tcp   filtered    globe
2003/tcp   filtered    cfingerd
2004/tcp   filtered    mailbox
2005/tcp   filtered    deslogin
2006/tcp   filtered    invokator
2007/tcp   filtered    dectalk
2008/tcp   filtered    conf
2009/tcp   filtered    news
2010/tcp   filtered    search
2011/tcp   filtered    raid-cc
2012/tcp   filtered    ttyinfo
2013/tcp   filtered    raid-am
2014/tcp   filtered    troff
2015/tcp   filtered    cypress
2016/tcp   filtered    bootserver
2017/tcp   filtered    cypress-stat
2018/tcp   filtered    terminaldb
2019/tcp   filtered    whosockami
2020/tcp   filtered    xinupageserver
2021/tcp   filtered    servexec
```

```
2022/tcp   filtered    down
2023/tcp   filtered    xinuexpansion3
2024/tcp   filtered    xinuexpansion4
2025/tcp   filtered    ellpack
2026/tcp   filtered    scrabble
2027/tcp   filtered    shadowserver
2028/tcp   filtered    submitserver
2030/tcp   filtered    device2
2032/tcp   filtered    blackboard
2033/tcp   filtered    glogger
2034/tcp   filtered    scoremgr
2035/tcp   filtered    imsldoc
2038/tcp   filtered    objectmanager
2040/tcp   filtered    lam
2041/tcp   filtered    interbase
2042/tcp   filtered    isis
2043/tcp   filtered    isis-bcast
2044/tcp   filtered    rimsl
2045/tcp   filtered    cdfunc
2046/tcp   filtered    sdfunc
2047/tcp   filtered    dls
2048/tcp   filtered    dls-monitor
2049/tcp   filtered    nfs
2053/tcp   filtered    knetd
5050/tcp   open        mmcc
5800/tcp   filtered    vnc-http
5900/tcp   filtered    vnc
6000/tcp   filtered    X11
6001/tcp   filtered    X11:1
6002/tcp   filtered    X11:2
6003/tcp   filtered    X11:3
6004/tcp   filtered    X11:4
6005/tcp   filtered    X11:5
6006/tcp   filtered    X11:6
6007/tcp   filtered    X11:7
6008/tcp   filtered    X11:8
6009/tcp   filtered    X11:9
6050/tcp   filtered    arcserve
7070/tcp   open        realserver
8080/tcp   open        http-proxy
9090/tcp   open        zeus-admin
13722/tcp  open        VeritasNetbackup
13782/tcp  open        VeritasNetbackup
13783/tcp  open        VeritasNetbackup
Remote operating system guess: Solaris 8 early access beta through actual release
Uptime 111.359 days (since Sat Dec  7 12:23:52 2002)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=151956 (Good luck!)
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 66 seconds
```

## 7.2 *Full Nessus Report*

```
Nessus Scan Report
------------------

SUMMARY

 - Number of hosts which were alive during the test : 1
 - Number of security holes found : 79
 - Number of security warnings found : 33
 - Number of security notes found : 58

TESTED HOSTS

 giacmain.example.org (Security holes found)

DETAILS

+ giacmain.example.org :
 . List of open ports :
   o smtp (25/tcp) (Security notes found)
   o ssh (22/tcp) (Security notes found)
   o ftp (21/tcp) (Security warnings found)
   o domain (53/tcp) (Security warnings found)
   o http (80/tcp) (Security hole found)
   o auth (113/tcp) (Security warnings found)
   o https (443/tcp) (Security hole found)
   o submission (587/tcp) (Security notes found)
   o unknown (3030/tcp) (Security notes found)
   o unknown (4040/tcp) (Security notes found)
   o mmcc (5050/tcp)
   o unknown (5222/tcp) (Security notes found)
   o unknown (5269/tcp) (Security notes found)
   o vnc-http-1 (5901/tcp) (Security notes found)
   o unknown (7070/tcp) (Security hole found)
   o unknown (7649/tcp) (Security notes found)
   o unknown (7648/tcp)
   o unknown (7802/tcp)
   o unknown (7878/tcp)
   o unknown (8001/tcp) (Security notes found)
   o unknown (8000/tcp) (Security notes found)
   o http-proxy (8080/tcp) (Security notes found)
   o zeus-admin (9090/tcp)
   o unknown (10001/tcp)
   o bpjava-msvc (13722/tcp) (Security notes found)
   o bpcd (13782/tcp) (Security notes found)
   o vopied (13783/tcp) (Security notes found)
   o domain (53/udp) (Security notes found)
   o general/udp (Security notes found)
   o general/tcp (Security hole found)

 . Information found on port smtp (25/tcp)

    An SMTP server is running on this port
    Here is its banner :
    220 giacmain.example.com ESMTP Sendmail 8.12.8/8.12.8; Fri, 28 Mar 2003
     20:17:46 -0800 (PST)

 . Information found on port smtp (25/tcp)

    Remote SMTP server banner :
    220 giacmain.example.com ESMTP Sendmail 8.12.8/8.12.8; Fri, 28 Mar 2003
     20:18:38 -0800 (PST)

    This is probably: Sendmail version 8.12.8

 . Information found on port smtp (25/tcp)

    smtpscan was not able to reliably identify this server. It might be:
    Sendmail 8.11.6
    Sendmail 8.12.2-8.12.5:
    Sendmail 8.11.6,8.12.3,8.12.5:
    The fingerprint differs from these known signatures on 2 point(s)

    If you known precisely what it is, please send this fingerprint
    to the Nessus team or Julien Bordet <zejames@greyhats.org>:
    :250:501:501:250:553:553:250:214:252:502:502:502:502:250:250

 . Information found on port smtp (25/tcp)

    Nessus sent several emails containing the EICAR
    test strings in them to the postmaster of
    the remote SMTP server.

    The EICAR test string is a fake virus which
    triggers anti-viruses, in order to make sure
    they run.

    Nessus attempted to e-mail this string five times,
    with different codings each time, in order to attempt
    to fool the remote anti-virus (if any).

    If there is an antivirus filter, these messages should
    all be blocked.

    *** To determine if the remote host is vulnerable, see
    *** if any mail arrived to the postmaster of this host

    Solution: Install an antivirus / upgrade it

    Reference : http://online.securityfocus.com/archive/1/256619
    Reference : http://online.securityfocus.com/archive/1/44301
    Reference : http://online.securityfocus.com/links/188

    Risk factor : Low

 . Information found on port ssh (22/tcp)

    An ssh server is running on this port

 . Information found on port ssh (22/tcp)

    Remote SSH version : SSH-2.0-OpenSSH_3.5p1

 . Information found on port ssh (22/tcp)

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

      . 1.99
      . 2.0

 . Warning found on port ftp (21/tcp)

    This FTP service allows anonymous logins. If you do not
     want to share data with anyone you do not know, then you should deactivate
     the anonymous account, since it can only cause troubles.
     Under most Unix system, doing :
      echo ftp >> /etc/ftpusers
     will correct this.

     Risk factor : Low
    CVE : CAN-1999-0497

 . Information found on port ftp (21/tcp)

    An FTP server is running on this port.
    Here is its banner :
    220-=(<*>)=-.:. (( Welcome to PureFTPd 1.0.12 )) .:.-=(<*>)=-

 . Information found on port ftp (21/tcp)
```

```
   Remote FTP server banner :
   220-=(<*>)=-.:. (( Welcome to PureFTPd 1.0.12 )) .:.-=(<*>)=-

. 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.

   See also : http://www.cert.org/advisories/CA-1997-22.html

   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
   CVE : CVE-1999-0024
   BID : 678

. Information found on port domain (53/tcp)

   The remote bind version is : 9.2.2

. Information found on port domain (53/tcp)

   A DNS server is running on this port. If you
   do not use it, disable it.

   Risk factor : Low

. Information found on port http (80/tcp)

   A web server is running on this port

. Information found on port http (80/tcp)

   The remote web server type is :

   Apache/2.0.44 (Unix) mod_ssl/2.0.44 OpenSSL/0.9.6g

   Solution : You can set the directive 'ServerTokens Prod' to limit
   the information emanating from the server in its response headers.

. Warning found on port auth (113/tcp)

   The 'ident' service provides sensitive information
   to potential attackers. It mainly says which accounts are running which
   services. This helps attackers to focus on valuable services [those
   owned by root]. If you don't use this service, disable it.

   Risk factor : Low

   Solution : comment out the 'auth' or 'ident' line in /etc/inetd.conf
   CVE : CAN-1999-0629

. Information found on port auth (113/tcp)

   An unknown service is running on this port.
   It is usually reserved for AUTH

. Information found on port auth (113/tcp)
```

```
    An Auth/ident server seems to be running on this port

. Warning found on port https (443/tcp)

    The SSLv2 server offers 5 strong ciphers, but also
    0 medium strength and 2 weak "export class" ciphers.
    The weak/medium ciphers may be chosen by an export-grade
    or badly configured client software. They only offer a
    limited protection against a brute force attack

    Solution: disable those ciphers and upgrade your client
    software if necessary

 . Warning found on port https (443/tcp)

    It seems that your web server rejects requests
    from Nessus. It is probably protected by a reverse proxy.

    Risk factor : None

    Solution : change your configuration
               if your tests to be accurate

. Information found on port https (443/tcp)

    A TLSv1 server answered on this port

 . Information found on port https (443/tcp)

    A web server is running on this port through SSL

 . Information found on port https (443/tcp)

    Here is the list of available SSLv2 ciphers:
    RC4-MD5
    EXP-RC4-MD5
    RC2-CBC-MD5
    EXP-RC2-CBC-MD5
    DES-CBC-MD5
    DES-CBC3-MD5
    RC4-64-MD5

 . Information found on port https (443/tcp)

    This TLSv1 server also accepts SSLv2 connections.
    This TLSv1 server also accepts SSLv3 connections.

 . Information found on port https (443/tcp)

    The remote web server type is :

    Apache/2.0.44 (Unix) mod_ssl/2.0.44 OpenSSL/0.9.6g

    Solution : You can set the directive 'ServerTokens Prod' to limit
    the information emanating from the server in its response headers.

 . Information found on port submission (587/tcp)

    An SMTP server is running on this port
    Here is its banner :
    220 giacmain.example.com ESMTP Sendmail 8.12.8/8.12.8; Fri, 28 Mar 2003
     20:17:59 -0800 (PST)

 . Information found on port submission (587/tcp)

    Remote SMTP server banner :
    220 giacmain.example.com ESMTP Sendmail 8.12.8/8.12.8; Fri, 28 Mar 2003
     20:18:47 -0800 (PST)

    This is probably: Sendmail version 8.12.8
 . Information found on port submission (587/tcp)

    smtpscan was not able to reliably identify this server. It might be:
    Sendmail 8.11.6
    Sendmail 8.12.2-8.12.5:
    Sendmail 8.11.6,8.12.3,8.12.5:
    The fingerprint differs from these known signatures on 2 point(s)

    If you known precisely what it is, please send this fingerprint
    to the Nessus team or Julien Bordet <zejames@greyhats.org>:
    :250:501:501:250:553:553:250:214:252:502:502:502:502:250:250

 . Information found on port submission (587/tcp)

    For some reason, we could not send the EICAR test string to this MTA

. Information found on port unknown (5269/tcp)

    The service closed the connection after 0 seconds without sending any data
    It might be protected by some TCP wrapper

 . Information found on port vnc-http-1 (5901/tcp)

    The service closed the connection after 0 seconds without sending any data
    It might be protected by some TCP wrapper

 . Vulnerability found on port unknown (7070/tcp) :

    The remote Real Server discloses the content of its
    memory when issued the request :

     GET /admin/includes/

    This information may be used by an attacker to obtain
    administrative control on this server, or to gain
    more knowledge about it.

    Solution : See http://service.real.com/help/faq/security/memory.html

    Risk factor : High
    CVE : CVE-2000-1181
    BID : 1957

 . Information found on port unknown (7070/tcp)

    A web server is running on this port

 . Information found on port unknown (7649/tcp)

    The service closed the connection after 0 seconds without sending any data
    It might be protected by some TCP wrapper

. Information found on port http-proxy (8080/tcp)

    A web server is running on this port

 . Information found on port bpjava-msvc (13722/tcp)

    The service closed the connection after 1 seconds without sending any data
    It might be protected by some TCP wrapper

 . Information found on port bpcd (13782/tcp)

    The service closed the connection after 1 seconds without sending any data
    It might be protected by some TCP wrapper

 . Information found on port vopied (13783/tcp)

    The service closed the connection after 1 seconds without sending any data
    It might be protected by some TCP wrapper
 . Information found on port domain (53/udp)

    A DNS server is running on this port. If you
    do not use it, disable it.

    Risk factor : Low

 . Information found on port general/tcp

    Remote OS guess : Solaris 8 early access beta through actual release

    CVE : CAN-1999-0454

------------------------------------------------------
This file was generated by the Nessus Security Scanner
```

# 8 Selected References and Recommended Reading

## 8.1 Security Information

- *CERT Coordination Center*, <http://www.cert.org>
- "The Solaris Security FAQ" by Peter Baer Gavin, <http://www.itworld.com/Comp/2377/security-faq>
- *The Art of Deception* by Kevin D. Mitnick and William L. Simon
- *Solaris Security Step by Step Version 2.0* by Hal Pomeranz
- *SANS Institute – Computer Security Education and Information Security Training*, <http://sans.org>
- "Defense In-Depth on a Solaris 2.X System" by Mark Strong, <http://www.sans.org/rr/unix/solaris_2x.php>
- "Secure BIND Template Version 3.7" by Rob Thomas, <http://www.cymru.com/Documents/secure-bind-template.html>

## 8.2 Security Tools

- Ifstatus, <http://www.cymru.com/Tools>
- John the Ripper, <http://www.openwall.com/john/>
- Logwatch, <http://www.logwatch.org/>
- Nessus, <http://www.nessus.org>
- Nmap, <http://www.insecure.org/nmap/>
- Noshell, <http://www.cert.org/security-improvement/implementations/i049.02.html>
- Npassword, <http://www.utexas.edu/cc/unix/software/npasswd/>
- OpenSSH, <http://www.openssh.com/>
- Sudo, <http://www.courtesan.com/sudo/>
- Tripwire, <http://www.tripwire.org> and <http://www.frenchfries.net/paul/tripwire/>