GIAC ENTERPRISES LTD.

Host Security Audit

GCUX Practical Assignment

By jonas Okwara

10/7/01

Table Of Contents

Executive summary:
This is an eight – week security audit of GIAC Enterprises LTD., involving two Sun enterprise 4500 servers. The auditors employed the TARA host scanning tool and ran commands to verify system configuration . They also interacted with the system and database administrators for more information.

The two machines, focus of the audit run Solaris 2.7 . One of them Hannibal, hosts an Apache web server , and the other Sparta, is an Oracle 8i database server . In addition to an E – commerce hosted on Hannibal, field marketing staff nation wide, log in to the Oracle database and download purchase orders. This means the very nature of the business of GIAC requires remote interaction with these machines, which is a genuine security concern.

The reports of the security audit found varying levels of vulnerability on the two hosts. They were for instance not configured as dedicated servers. Many services were found to be enabled in the /etc/inetd.conf file outside the control of the tcp- wrappers daemon. The tcpd program would have provided access restriction but it was not installed.

Also enabled were the Sendmail daemon (smtpd) and the Simple network management protocol daemon (snmpd) spawned as boot scripts in the /etc/rc2.d and /etc/rc3.d directories respectively. Hannibal and Sparta were also unsecured against buffer overflow attacks. Morever the users were running their cron jobs as root and several files had the SETUID and the SETGID bit enabled on them. The Apache web server had third party Cgi scripts running

Filesystem integrity was also found to be vulnerable. In addition, the system administrators who installed and configured the Solaris operating system on the machines did not do an initial checksum of system binaries to enable them detect malicious alterations in the filesystem.

Login authentication at GIAC Enterprises is by the SSH protocol, which also encrypts the sessions, although the marketing staff thinks the login routines of the SSH protocol are too technical and often task the patience of its field staff. Password aging is unenforced because there is no such policy. Equally vulnerable is the root account. The system and database administrators often do their work with root and no effort was made to install the Sudo program to delegate root functions.

Scheduled backups exist for disaster recovery but there is no policy of off site storage of duplicate backup media which is wrong. Also the Solaris platform which hosts the Apache web server should have been clustered to ensure that the web site will still be on line should there be a denial of service attack which sabotages the server.

The auditors also learnt that the system Administrators responsible for these two hosts are good at routine system administration but do not have the requisite training and therefore lack the skills to handle unix system security . The result is poor configuration of the Solaris Operating environment. The auditors recommended the SANS course for the Systems Administrators and measures to tighten security on the severs to ready them for the projected growth in business and the growing threat from the internet.

Analyses of Findings:

1)**Operating System issues**:
Sun Microsystems's Solaris operating System has a higher share in the Unix server market . Its platforms have over the years been a test bed for most of the security tools for unix , a development the hacker community is also aware of. It is likely therefore that most of the "Bad People" out there earned their spurs breaking into Sun machines. It is also true that although many of the security holes in Solaris Operating system have been plugged , it still remains a platform more familiar to hackers. This means there is a measure of risk involved in running Solaris which invariably calls for more stringent security measures. Being aware of this information is a first step for a security administrator.

Again unlike Linux , Solaris does not come with all the array of security freeware tools like tcpwrappers, tripwire, sudo, ipchains and IsOf. Also Solaris does not come with commands like chkconfig neither does it have a C compiler to enable skilled administrators build some of the freeware available on the internet. It takes good skills to download and build freely available software. This, the system administrators admitted, was a major problem they faced in their efforts to secure the two unix servers. Besides they admitted having difficulty compiling the gcc available as a freeware. However the admins admitted they were up - to - date with the operating system patches the company has a service contract with Sun which gives it access to an extended set of patches and a complete database of patch information via the world wide web **(www.sun.com)** or anonymous ftp. OS commands like : *showrev –p* or *patchadd –p.* displays information about patches in the system. Moreover the *uname -a* command shows among other things, the patch base code, a hyphen and a number that represents the patch revision number . The output of these commands could compare with any revision updates from Sun to verify whether the system is up –to-date. It is recommended that companies purchase this service as part of their contract with Sun to have access to security patches and other recommended patches at the SunSolve database of patches via the above URL.

2)Network Services issues

By interacting with the two servers , through running relevant operating system commands, opening and seeing system configuration files and analyzing the reports of the TARA host scanning tool, (*Appendix A*) there was little doubt that much still needs to be done to secure the two servers against the threats from the internet.

A look at the */etc/inetd.conf* file and the */etc/rc2.d , /etc/rc3.d* directories showed that the two servers were not configured to dedicate them to the specific roles they were to play in business . Several network services were enabled on Hannibal and Sparta which is wrong. These included those started from the *inetd.conf* file and boot scripts in the */etc/rc** files .

Below are some of the services running on the hosts. Their vulnerabilities and how this can be fixed or controlled is also discussed. Analysis of the Tcp-Wrappers daemon and how it can control access to these network services is also treated.

(FTP TCP port 20 & 21)

The file transfer protocol is enabled on the two servers and used to transfer patches between them . This is wrong because ftp requires login with a username and password which are transmitted across the network unencrypted. *(see Appendix G)* With a username and password, a malicious attacker can compromise the servers and even upload files to the web server. FTP should therefore be disabled in the */etc/inetd.conf* file by commenting it out and sending the " kill -HUP (process id) Signal to the inetd daemon.

Alternatively it can be brought under the control of the tcp – wrappers daemon, which means access to the ftp program is limited to only ip addresses in the /etc/hosts.allow file referenced by the tcpd. However for a dedicated web server, it might not be advisable to enable the ftp daemon at all because it could be used to upload malicious code to directories like the *cgi/bin.*

. **Telnet TCP Port 23):**

Telnet which is a remote login service was also found to be enabled on Hannibal and Sparta. Unfortunately telnet transmits usernames and passwords in plaintext over the internet where a malicious third party can intercept them. Additionally an entire telnet session is readable by a network snooper. Telnet is also one of the most commonly trojaned programs in unix and should therefore be disabled on any mission critical server.
It is disabled by commenting it out in the */etc/inetd.conf* file and sending the inetd daemon the " kill –HUP (process id) " signal to re-read its configuration. As in the case of the ftpd, telnetd can also be controlled by the tcpd daemon. Alternatively installing and running the client program *(ssh)* of the Secure Shell is better than enabling the insecure telnet service. Secure Shell integrates well with tcpwrappers if properly configured.

(Smtpd   TCP port 25):
GIAC enterprises has sendmail enabled on its servers which is also wrong considering sendmail has been the source of numerous security breaches on unix systems. Earlier versions of sendmail for instance allowed mail to be sent directly to any file on the system including the */etc/passwd file*. Sendmail also allows trusted users who are permitted to forge and deliver mail to the local system. Compiled in " debug mode ", sendmail would allow outsiders unrestricted access to the system it is running on. It can also vulnerable to buffer overflow attacks and so should not be enabled on servers running business applications.

Sendmail is turned off in the /etc/rc2.d directory by renaming it with the commands :

```
# cd  /etc/rc2.d
 # mv S88sendmail  Off_S88sendmail
```

Fingerd TCP Port 79):
Similarly, the finger daemon was found to be running on the two servers, which is dangerous . If the finger command is executed with no arguments the program prints the username, full name, location, login time and office telephone number of every user currently logged into the two machines .

If the command is executed with a name argument, the program searches through the */etc/passwd* file and prints detailed information for that user including first name , last name the home directory, the shell login time and the IP address of the username that matches the specified argument.

Some versions of the fingerd even print the contents of the .plan and .project files in a persons home directory. Enabling the finger daemon on mission critical servers is improper because the information revealed by fingerd could be used as the basis for social engineering attack. (*Appendix I*) Fingerd is disabled by commenting it out in */etc/inetd.conf* file and sending the inetd daemon the appropriate signal to re-read its configuration.

(Tftpd UDP Port 69)
The Tftp is a UDP – based file –transfer program that provides no security . There is a set of files that the TFTP program is allowed to transmit from the computer and the program will transmit them to anybody on the internet. There was no need for this service on the two servers and should be disabled in the */etc/inetd.conf* file.

Snmpd UDP Port 161 & 162):

The simple network management protocol is designed to allow the remote management of devices on the network. It allows messages that monitor current status of the network and messages that change the status of network devices . however SNMP can be of great value to attackers. With carefully constructed snmp messages an attacker can learn the internal structure of a local area network, change network configurations and even shut down operations. The snmp server starts as a boot script in Solaris and should have been disabled to limit the information an attacker can learn about the GIAC network. It should be disabled by turning it off in the /etc/rc3.d with the following command :

```
# cd   /etc/rc3.d ,  mv  S76snmpdx   Off_snmpdx
```

The Tcp Wrappers Solution :

Tcpwrappers is a freeware application which provides finer access control mechanism for those network services mentioned above. Sadly though, it controls only those services originally controlled by the inetd daemon. At its basic configuration, tcpwrappers consists of the tcpd daemon, a hosts.allow file and hosts.deny file. Instead of turning off all the network services mentioned above, the ones enabled, are taken off the control of the inetd daemon and brought under the control of the tcpd daemon, which references the hosts.allow and hosts.deny files before granting access to those services it controls. For instance IP addresses specified in the /etc/hosts.allow file are given access to the specified service(s) in that file. Similarly, IP addresses in the /etc/hosts.deny file are denied access to service(s) specified in that file. The default Setup for these files is to deny access to all and then specify in */etc/hosts.allow*, those IP addresses which should be allowed access to specified services.. *[see Appendix K for Install and configuration details]*

The tcpwrapper application is a popular security tool for security administrators in addition to its finer access control mechanism, it also logs its activities to the syslog facility.

(2) The Filesystem integrity issue

The system disks in the two servers were properly partitioned , with each filesystem taking up a slice. In particular there was enough space for */usr* and */var* . The possibility of any denial of service attack by overwhelming these critical slices with third party binaries and logs is therefore remote. However There was no effort to obtain a checksum of the system binaries immediately following the installation of operating system.

This means system binaries could be altered , rootkits and Trojan horse programs could be inserted in the filesystem hierarchy without the systems administrators ever detecting the alteration. Programs like the pkgadd and cksum are native to Solaris and can be used in a shell script to alert administrators about changes in system binaries. Other freeware tools like tripwire are also handy for doing a similar audit of the filesystem particularly the */dev* directory, usually a hiding place of choice for rootkit programs. Administrators should be aware that Kernel and application patches do also create differenes in binary checksums. However a dutiful admin who places his system binaries and filesystem under constant watch will certainly detect a malicious alteration.

(3) Buffer Overlow and Core File issues

The two hosts were not configured to protect them from buffer overflow attacks considering the necessary entries in the */etc/system* files for the two servers were missing. Buffer overflows are an active attempt to manipulate the memory space of a running process in order to execute arbitrary code inserted by the attacker.

In Solaris 2.6 and later some of these exploits can be avoided by adding the following entries: *set noexec_user_stack = 1* in the */etc/system* file to make the system stack non-executable. (*see Appendix L)* Also with the *noexec_user_stack_log* set to one, the operating system logs programmatic attempts to execute code on the stack. This allows an intrusion detector to track unsuccessful exploit programs and the account which made the attempt.

Also buffer overflows are another way of triggering core dumps . These files are the remains of an aborted system process. They are the image of a memory allocated to a program and could contain sensitive information such as the */etc/shadow* file which could be used to attack the system. To prevent this security administrators must add the following entry below to the */etc/system* file to prevent the creation of core files.

```
  *   Set  various parameters to more reasonable values
      Set maxuprc    =  128
   Set  sys:coredumpsize   = 0
```

For security reasons, Solaris operating system will not write core files for processes with an effective ID that is different from the real ID . This means *SUID* and *SGID* programs will not create core files.

But if the system must dump core then the coreadm utility could be used . The coreadm utility ships with Solaris 7 and later for managing core files. With the coreadm utility , a system can be configured so that all process core files are placed in a single directory. This makes it easier to track problems by examining the core files in a specific directory whenever a Solaris process or daemon terminates abnormally.

Executing the *coreadm* command gives the default settings of the */etc/coreadm.conf* file. There are many options to the *coreadm* command. (See the man pages for more details). For instance the command below configures system wide parameters that add the executable file name and PID to the name of any potential core file that might be created in the /var/core directory.

Coreadm –g /var/core/core.%f.%p –e global

When created with the above configuration, global core files are created in permissions mode 600 and are owned by the superuser. Non privilged users cannot examine such files. It is always advisable to execute the *coreadm* command to confirm changes to the/etc/coreadm.conf file following any configuration.

(4 System Auditing and Logging issues:

Of equal importance are system logs. Logs files are an important building block of a secure system. They form a recorded history or audit trail of the computer's past. Logs can help to track problems or attacks. They can also assist in investigation or be what one needs to rebuild the system. However it is always a wise idea to have a loghost where system messages will be remotely logged. This is configured in the */etc/syslog.conf* **[See appendix L for configuration details.]** file where, syslogd could be configured to log messages remotely instead of the localhost. There was no attempt to enable any of these machines to remotely log their syslog messages in another host which is a major configuration flaw. The possibility therefore existed that an intruder who gains access to these servers would easily clear them of these critical logs to conceal his activity. The advantage of logging in multiple places is that it makes an attacker's attempts at erasing any evidence of his presence much more difficult. Without logs a forensics investigator will have great difficulty re-constructing a major security incident.

(5)Log and Network Time protocol issues:

From a security point of view, it is always proper for the servers to have accurate time stamps on all log files to enhance investigation of audit trails of a break in and effective prosecution of security incidents. A contrary situation will complicate or jeopardize uniformity and legal proceedings. This calls for time syncronisation in a network. GIAC should implement an Network Time Protocol (NTP) infrastructure involving Hannibal and Sparta in a peer – to – peer relationship. Both will sync time information from each other and in turn syncronize from a secondary internal ntp server. The secondary ntp server will sync from a primary source of time information from the internet. The Network Time Protocol (NTP) public domain software from the University of Delaware is included in the Solaris software starting with the Solaris 2.6 release. The xntpd daemon is a complete implementation of the version 3 standard, as defined by RFC 1305. However the NTP source code is available via ftp.udel.edu. It includes a GNU configure script. Once this is completed, the *make* and then *make install* utilities are executed. To intall the Opensource version for Solaris 2.6 and earlier, the parameter *dosynctodr should be set to 0* in the /etc/system file to avoid conflicts between NTP and the hardware clock. For the configuration of the NTP network specified above see (Appendix L).

(6) Cron/SetUid/SetGid issues

Some of the systems and database administrators were found by the TARA hosts scanner to be running their own scripts in the root's crontab. (*Appendix B*) This is a classic mistake in using the cron utility. The cron system is designed to run jobs as different users – particularly automated tasks which run as root.

There is always the great possibility that a malicious user will get unprivileged access to the script file by using the su program as root to become the owner of the script. From there the attacker can rlogin or ssh to the system where the scripts resides and alter the script to make himself a set- UID copy of the shell in some directory. Only root should be allowed to run the crontab and at commands. This can be accomplished with the following configuration:

```
# cd  /etc/cron.d
# rm –f  cron.deny   at .deny
#  echo  root   >   cron.allow ,   echo root > at.allow
#  chown   root:root   cron.allow   at.allow
#  chmod  400  cron.allow   at.allow
```

Also output of the TARA hosts scan showed files with SETUID and SETGID bit on them and owned by root. These files allow normal users to execute them as though they were the owner. Ordinarily a command like " passwd " which is setuid root, is important for normal users to change their password. However attackers may also use setuid or setgid features on a file to create backdoors to the system. One way of doing

this is by copying a system shell to a hidden location and adding the setuid bit which allows the attacker to execute the shell to gain elevated privileges (most often superuser).
Some of them owned by root are not required for the normal operation of the system and should be deleted. Periodic search for SETUID and SETGID files is recommended with the command below:

```
#  Find  / -type  f \ ( -perm –u+s  -o  -perm –g+s  \ )  -ls
```

<u>(7)User Account issues:</u>

<u>(a) The Superuser Account:</u>
The goal of most hackers is to obtain knowledge of the root account and its password or run SUID programs which execute with root privilege. The importance of the superuser has made its password a well guarded secret in most companies. At GIAC it was learnt that what ever existed as a password policy was never enforced.

The two servers have a total of 12 systems administrators, database administrators , web developers and marketers . each of these had an account in the systems. Because of the nature of their duties, the DBAs need root access which the unix admins willingly oblige. This is wrong since sudo could be used to assign to the DBAs the appropriate root commands and programs they need to do their work. But Sudo is not installed on the machines .

Giving out the root password to many people is another way of lowering the security hedge over the hosts since the passwords could fall into wrong hands. Although Sys Admins who quit their jobs always have their accounts deleted, in most companies, due to administrative lapses this is not immediate. A former employee who knows the vulnerabilities in his/her previous network, could end up as another member of the hacker community. With the root passwd he will always be a security threat to the systems. It is therefore preferable to delegate root functions with the Sudo program than allowing all and sundry to have the superuser password.

The two hosts have /etc/shadow files for password encryption and aging but password aging is not enforced.

Similarly, there were system default accounts that should be deleted or modified to strengthen security. Some of these system accounts listed in the /etc/passwd are not necessary for normal system operation. They include smtp , nuucp and listen etc. They have no shell listed and have the “ NP” string listed in the /etc/shadow file meaning they have no passwords. Some additional steps can be taken to add more security, like running the following command to lock the account:

```
# Passwd  –l  uucp
```

Some administrators however prefer to monitor these system accounts for abuse and use a security package called Titan which includes a shell replacement called *noshell*. When executed, *noshell* logs entries which allows administrators to track unauthorized use of system accounts.

(8) **Remote access/ Encryption issues** :

Remote access to the root account is disallowed through configurations in the /etc/default/login file and the /etc/default/su file which logs all attempts to use the superuser account. Normal remote logins on Hannibal, the database server is by the Secure Shell protocol through a public/private - key technology to initialize a secure connection. It also encrypts the sessions and securely forwards X11 and TCP/IP connections. The SSH2 (ie 2.4.0) is the version in use and it is advisable to be up to date on developments around the SSH2 technology to be aware of bugs and fixes. The problem is that the administration of SSH is contracted to the same company which manages GIAC's firewall and is also responsible for data encrption .

The marketing officers nation wide log in to the database server to download orders through a client application. This transaction is also encrypted even though they complain that SSH is too technical.

(9) Third party Software issues:

(a)Version of Apache: Sparta and Hannibal , the two Solaris servers host an Oracle database and Apache web server respectively as the only third party software running on the hosts. The flawed practice of allowing the Oracle DBAs access to the root password has already been discussed above ***(see User Account Issues above)*** along with the Sudo remedy. The version of Apache running on Hannibal is version 1.3.13 . It also has the Secure Sockets Layer (SSL) for strong authentication and end – to – end encryption at the network socket layer. Still it is always wise to be up to date with information on any bugs and its fixes. Access rights were found also to be properly configured. For instance access to the root directories were denied to all and the httpd runs as nobody. The Oracle database was also up to date on patches.

(b)Apache Cgi Scripts:

Third party cgi scripts were running on the Apache web server . The scripts had the UID and GID as the web server, which is the user “nobody”. “Nobody” usually does not own any files or directories . Cgi scripts can present security holes in two ways: They may intentionally or unintentionally leak information about the host which will help hackers break in. Moreover scripts that process remote user input such as the contents of a form or a searchable index command may be vulnerable to attacks in which the remote user tricks them into executing commands. They should be coded with security in mind and should be in the cgi-bin directory.

If they must be used , third party CGIs such as shopping cart programs or image manipulation programs must be tested in a controlled environment on an isolated network and monitored with tcpdump or some network sniffer. The sniffer will check to see if data is being served out by mail or other forms of communication. Open source CGIs must also be examined vigorously. For instance : How are the parameter values loaded into internal variables . Are they copied into fixed – length buffers ? System calls should also be looked at . If the script is written in Perl check for the magic characters in calls to open(). In addition examine the UID/GID execution bit set on the script.

(10)Backup/Restore issues:

The audit found GIAC to have a regular schedule of backup and restore , meaning that in normal circumstances any of the two systems could be brought up to its productive stage. However backup media were all found to be stored in – house . This is wrong practice because it will make it impossible for the company to recover from a catastrophic disaster like fire or flooding .

There are instances when information in backup tape media was found to have been erased probably due to poor handling or strong magnetic fields in storage rooms. Off – site storage of duplicate tape media is always a better solution. Backups are not only for disaster recovery but to compare OS files against a “gold” image whenever there is a security incident. GIAC has standby servers to handle any unexpected systems failure. This solution is good but does not minimize the downtime between system

failure and bringing up the standby server. A better solution will be to have a Sun clustering or Veritas clustering arrangement where the Apache web server will fail over to the next server should the online e-commerce server go down due to a denial of service attack or hardware failure. For an e-commerce site with an ambitious revenue target clustering the servers will be a better solution because downtime is minimized.

(11) Intrusion Detection

There is no intrusion detection system to alert the system administrators when there is a break in. Tools like *logchecker, tripwire* and *IsOf* are some of the popular freewares available. Also Nmap, the port scanner must be installed to monitor the ports. Also a dutiful administrator could write and schedule custom scripts to alert him when the hosts have been compromised or to detect changes in system files or binaries. The */etc/passwd* database must be monitored for unauthorized alterations or creation of user accounts. The root account should also be examined periodically for changes.

(12)Administrative practices:

An apparent lack of a security policy in GIAC Enterprises is the single most powerful factor militating against a secure computing environment. Most of the configuration flaws mentioned like password aging, access to the root password and DBA rights etc are issues which must be defined by a policy. Also administration of the firewall is by another company to which the job is contracted. The contract covers secure login authentication and data encryption .

A situation therefore exists whereby the contractors who manage the firewall and the systems Administrators who manage the servers see themselves as distinct from one another . Instead of complementing each other they see their jobs as mutually exclusive. Whereas the systems administrators were found to be great at routine systems administration tasks, they showed poor unix security admin skills to enable them eliminate those vulnerabilities discovered by the audit.

Conclusion/Recommendation:

Defending an enterprise with a firewall is a great idea. But better wisdom will also go for a host based solution which complements the firewall so that if there is a breach of the main defences the host servers will not be utterly vulnerable, at the mercy of the malicious intruder. This complementarity is lacking in the network under evaluation. A security policy would have spelt out these issues to show strong commitment to a secure computing environment. A strong commitment will also ensure administrators are better trained to acknowledge security loopholes and know what to do. The SANS unix classes on security is highly recommended in this regard.

To recap some of the recommendations in earlier chapters, the administrators should purge the boot directories of all unwanted services. Similarly, unnecessary services in the /etc/inetd.conf file must be disabled or deleted , to allow the Apache web server in particular run as a dedicated httpd server. The use of telnet for remote login and ftp for transferring patches between the servers should be discontinued. A better solution will be to install the Tcpwrappers on these machines and limit access to only authorized IP addresses in the /etc/hosts.allow file. The SSH protocol should also be integrated to this configuration to encrypt the sessions. The use of root's crontab by normal users should also be stopped, and the /etc/cron.d/cron.allow file should contain only root. Similar settings should go for the /etc/cron.d/at.allow file. Unnecessary SetUid and SetGid files should also be deleted.

The admins should do a binary audit of the filesystem and have a proper checksum of these binaries. Periodic checksums should be compared with this initial one in a shell script to alert the systems admins to a likely alteration in the system binaries, although they should be aware that patches could also create differences in the checksum.
Cgi scripts must be in the cgi-bin directory and the admins must be aware of scripts that give out data by mail or other forms of communication.

In all , the job of the security administrator is to firmly secure the servers
With the right system configurations so that when they ever try, hackers will always be frustrated because a dutiful and knowledgeable administrator had done the right things.

Note on priority levels:

This is a prioritized chart on handling the fixes. The priority is on the risk level as indicated by high or medium. It is however subjective and only meant as a guide to solving the issues raised.

Prioritized List of Vulnerabilities , Fixes , Personnel and Cost

1)Vulnerable issues	***Risk Level***	***Solution***	***Personnel***	***Cost***
2) Unnecessary services in inetd.conf file	***high***	***Disable them all in /etc/inetd.conf or wrap them in tcpd.***	***2***	***$50 /hr.***
3) Unnecessary boot scripts spawned from /etc/rc*.d.	***High***	***Purge the boot Directories of unnecessary services or Rename them with the mv command.***	***2***	***$50/hr***
4) Apache configuration	***High***	****Run Apache as a dedicated server .*** . ***Keep Apache up to date on patches.***	***2***	***$50 /hr***
5) Filesystem integrity (/usr) directory.	***High***	****Do a checksum of OS binaries under /usr***	***2***	***$50***
6) Filesystem integrity (/dev) directory.	***High***	****Do checksum of files in /dev*** ****Install tripwire to alert on changes in system binaries.***	***2***	***$50***
7) Memory attacks (buffer overflow)	***High***	***Add the following in /etc/system:*** ***set noexec_user_stack = 1*** .	***1***	***$50/hr @ 1 hr***
8) memory	***High***	***Set the following***	***1***	***$50/hr @ 1 hr***

attacks (core dumps)		parameters in /etc/system: Set maxuprc = 128 Set sys:coredumpsize = 0 *Use coreadm utility to specify a directory for core dumps.		
9) Memory attacks	High	Add another line to /etc/system file To log attempts at stack attacks.	1	$50/hr @ 2hrs
10) System logs	Medium	*Install another server *Configure it as a log host, Set syslogd to Messages to loghost.	2	$50/hr @ 2hrs.
11) Access to relevant network services	High	Disable all unnecessary services or Install tcpwrappers Daemon.	2	$50/hr @ 3 hrs
12) SETUID & SETGID files	High	Remove unnecessary root owned SUID & SGID files. Regularly Search for files and remove as required.	3	same
13) security advisory	Medium	Put appropriate Message in /etc/issue file	1	$50/hr @ 1hr.

14) Access to cron facility	High	Only root should be allowed to run cron and at jobs	1	$50/hr @ 1hr
15) superuser password	High	*install sudo and assign appropriate root tasks to dbas and sys admins	2	$50/hr @ 2hrs
16) Disaster Recovery (backup)	High	Store duplicate backup Media off-site	3	$50/hr every week.
17) Disaster Recovery (standby server)	High	*Install another Apache server and cluster two or more servers for fail over.	3	$50/hr @ 5hrs

18) intrusion	High	*install	3	$50/hr @ 4hrs

detection		logcheck on loghost to mail to administrator, clues of possible attacks *install port scanner like Nmac to periodically probe the two hosts.		
19) Password Aging	Medium	Enforce password aging On two systems	2	$50/hr periodically.
20)Security Policy	Medium	*GIAC should define its security policy *It should harmonize its firewall operations with those of the sys admins	---	---
21)Requisite skills for Sys Admins	High	SANS unix classes Highly recommended To sys admins	3	$3000 for each administrator.
22) Cgi script issues	high	Use a sniffer to check for attempts to “sneak” out data by mail or other forms of communication Cgi scripts should be in cgi-bin directory to enable monitoring. * Run CGIs as “nobody”..	4	$50/hr @ 4hrs
23)	High	*Allow only	-	Same

Cron issues		***root to run jobs in cron and at.*** **** Remove other users in cron.allow file***		
24)Network Time protocol and log files	***medium***	***Set up an ntp Infrastructure:*** ****Sync Hannibal & Sparta with each other.*** ****Sync both with an internal secondary time server.*** ****Sync secondary server with primary time source on internet.***	***5***	***same***

Appendix A

Edited Reports of TARA Scan on host Hannibal

```
Security scripts *** 2.0.9 ARC, 1999.0907.2100 ***
Mon Aug 27 12:48:40 EDT 2001
12:48> Beginning security report for Hannibal

( 1 )  Checking accounts from /etc/passwd.

--WARN-- [acc001w] Login ID adm is disabled, but still has
a valid shell.
--WARN-- [acc005w] Login ID adm is disabled, but has a
'cron' file or cron
         entries.
--WARN-- [acc001w] Login ID bin is disabled, but still has
a valid shell.
--WARN-- [acc001w] Login ID daemon is disabled, but still
has a valid shell.
--WARN-- [acc001w] Login ID listen is disabled, but still
has a valid shell.
--WARN-- [acc001w] Login ID lp is disabled, but still has a
valid shell.
--WARN-- [acc005w] Login ID lp is disabled, but has a
'cron' file or cron
         entries.
--WARN-- [acc001w] Login ID noaccess is disabled, but still
has a valid shell.
--WARN-- [acc001w] Login ID nobody4 is disabled, but still
has a valid shell.
--WARN-- [acc001w] Login ID sys is disabled, but still has
a valid shell.
--WARN-- [acc005w] Login ID sys is disabled, but has a
'cron' file or cron
         entries.
--WARN-- [acc001w] Login ID uucp is disabled, but still has
a valid shell.
--WARN-- [acc005w] Login ID uucp is disabled, but has a
'cron' file or cron
         entries.
--WARN-- [acc006w] Login ID adm's home directory (/var/adm)
has group `sys'
         .
```

Appendix B

(Continuation of Reports of TARA scan)

(2) **<u>Performing check of /etc/hosts.equiv and .rhosts files...</u>**

```
--WARN-- [rcmd006w] User shawn .rhosts file has group
`sysadmin' and world
    read


--WARN-- [cron001w] cron entry for lp does not use full
pathname:
--WARN-- [cron001w] cron entry for lp does not use full
pathname:
--WARN-- [cron001w] cron entry for root does not use full
pathname:
--WARN-- [cron001w] cron entry for root does not use full
pathname:

--WARN-- [cron001w] cron entry for lp does not use full
pathname: --WARN-- [cron001w] cron entry for lp does not
use full pathname:
--WARN-- [cron001w] cron entry for root does not use full
pathname:
--WARN-- [cron001w] cron entry for root does not use full
pathname:



--WARN-- [cron002] cron entry for root uses
         `/export/home/gwc/Daily/setup_daily' which is not
owned by root


--WARN-- [cron002] cron entry for root uses
         `/export/home/gwc/Daily/setup_daily' which
contains
         `/export/home/gwc' which is not owned by root
(owned by gwc).

--WARN-- [cron002] cron entry for root uses
         `/export/home/gwc/Daily/setup_daily' which
contains
         `/export/home/gwc/Daily' which is not owned by
root (owned by gwc).
```

Appendix C

```
--WARN-- [cron002] cron entry for root uses
         `/export/home/gwc/Daily/setup_daily' which is not
owned by root
         (owned by gwc).
```

(4)**Performing NFS exports check...**

```
      none
```

(5)**Performing checks for SunOS...**

```
--WARN-- [no-id] The PROM monitor is not in secure mode.
--WARN-- [misc008w] NFS port checking disabled in kernel.

# Running './scripts/check_sendmail'...
```

(5) **Checking setuid executables...**

```
--FAIL-- [fsys001f] File /etc/lp/alerts/printer is a setuid
script:
-r-sr-xr-x   1 lp       lp           203 Dec 16  1999
/etc/lp/alerts/printer
--WARN-- [fsys002w] setuid program /usr/bin/nispasswd has
relative pathnames.
--WARN-- [fsys002w] setuid program /usr/bin/passwd has relative pathnames.

--WARN-- [fsys002w] setuid program /usr/bin/yppasswd has
relative pathnames.
--WARN-- [fsys002w] setuid program
/usr/lib/fs/ufs/ufsrestore has relative
         pathnames.
--WARN-- [fsys002w] setuid program
/usr/openwin/bin/kcms_calibrate has
         relative pathnames.
--WARN-- [fsys002w] setuid program
/usr/openwin/bin/kcms_configure has
         relative pathnames.
--WARN-- [fsys002w] setuid program
/usr/openwin/bin/sparcv9/kcms_configure has
         relative pathnames.
--WARN-- [fsys002w] setuid program /usr/openwin/bin/sys-
suspend has relative
         pathnames.
```

Appendix D

```
--WARN-- [fsys002w] setuid program /usr/sbin/static/rcp has
relative
         pathnames.

--CONFIG-- [fsys003c] No setuid list... listing all setuid
files
---s--x--x root     bin      /usr/lib/pt_chmod
---s--x--x root     uucp     /usr/bin/ct

---s--x--x uucp     uucp     /usr/bin/cu
---s--x--x uucp     uucp     /usr/bin/uucp
---s--x--x uucp     uucp     /usr/bin/uuglist
---s--x--x uucp     uucp     /usr/bin/uuname
---s--x--x uucp     uucp     /usr/bin/uustat
---s--x--x uucp     uucp     /usr/bin/uux
---s--x--x uucp     uucp     /usr/lib/uucp/remote.unknown
---s--x--x uucp     uucp     /usr/lib/uucp/uucico
---s--x--x uucp     uucp     /usr/lib/uucp/uusched
---s--x--x uucp     uucp     /usr/lib/uucp/uuxqt
-r-s--x--x root     bin      /usr/lib/lp/bin/netpr
-r-sr-sr-x root     daemon   /usr/dt/bin/sdtcm_convert
-r-sr-sr-x root     sys      /usr/bin/nispasswd
-r-sr-sr-x root     sys      /usr/bin/passwd
-r-sr-sr-x root     sys      /usr/bin/yppasswd
-r-sr-sr-x root     sys      /usr/dt/bin/dtaction
-r-sr-xr-x lp       lp       /etc/lp/alerts/printer
-r-sr-xr-x root     bin      /usr/bin/crontab
-r-sr-xr-x root     bin      /usr/bin/eject
-r-sr-xr-x root     bin      /usr/bin/fdformat
-r-sr-xr-x root     bin      /usr/bin/login
-r-sr-xr-x root     bin      /usr/bin/pfexec
-r-sr-xr-x root     bin      /usr/bin/rcp
-r-sr-xr-x root     bin      /usr/bin/rdist
-r-sr-xr-x root     bin      /usr/bin/rlogin
-r-sr-xr-x root     bin      /usr/bin/rmformat
-r-sr-xr-x root     bin      /usr/bin/rsh
-r-sr-xr-x root     bin      /usr/bin/sparcv7/uptime
-r-sr-xr-x root     bin      /usr/bin/sparcv7/w
-r-sr-xr-x root     bin      /usr/bin/sparcv9/uptime
-r-sr-xr-x root     bin      /usr/bin/sparcv9/w
-r-sr-xr-x root     bin      /usr/bin/volcheck
-r-sr-xr-x root     bin      /usr/bin/volrmmount
-r-sr-xr-x root     bin      /usr/dt/bin/dtappgather
-r-sr-xr-x root     bin      /usr/dt/bin/dtprintinfo
-r-sr-xr-x root     bin      /usr/dt/bin/dtsession
-r-sr-xr-x root     bin      /usr/lib/fs/ufs/quota
```

Appendix E:

```
-r-sr-xr-x root     bin      /usr/lib/fs/ufs/ufsdump
-r-sr-xr-x root     bin      /usr/lib/fs/ufs/ufsrestore
-r-sr-xr-x root     bin      /usr/lib/sendmail
-r-sr-xr-x root     bin      /usr/lib/utmp_update
-r-sr-xr-x root     bin      /usr/sbin/sparcv9/whodo
-r-sr-xr-x root     bin      /usr/sbin/static/rcp
-r-sr-xr-x root     bin      /usr/sbin/traceroute
-r-sr-xr-x root     sys      /usr/bin/chkey
-r-sr-xr-x root     sys      /usr/bin/sparcv7/ps
-r-sr-xr-x root     sys      /usr/bin/sparcv9/ps
-r-sr-xr-x root     sys      /usr/bin/su
-r-sr-xr-x root     sys      /usr/ucb/sparcv7/ps
-r-sr-xr-x root     sys      /usr/ucb/sparcv9/ps
-rwsr-sr-x root     bin
/usr/openwin/bin/kcms_calibrate
-rwsr-sr-x root     bin
/usr/openwin/bin/kcms_configure
-rwsr-sr-x root     bin
/usr/openwin/bin/sparcv9/kcms_configure
-rwsr-xr-x root     adm      /usr/lib/acct/accton
-rwsr-xr-x root     bin      /usr/openwin/bin/sys-suspend
-rwsr-xr-x root     bin      /usr/openwin/bin/xlock
-rwsr-xr-x root     bin      /usr/openwin/lib/mkcookie
-rwsr-xr-x root     bin      /usr/sbin/allocate
```

(6) **Checking setgid executables...**

```
--CONFIG-- [fsys003c] No setgid list... listing all setgid
files

# Checking unusual file names...

# Looking for unusual device files...

# Checking symbolic links...

# Checking for writable directories...
--INFO-- [fsys008i] The following directories are world
writable:
/var/dt/dtpower/schemes/
/var/dt/tmp/
/var/mail/
/var/preserve/
```

Appendix F

```
configuration file for inetd(1M).  See inetd.conf(4)
# for other services, see the Solaris System Administration
Guide uide.
# You must verify that a service supports IPv6 before
specifying <proto> as
# The remote shell server (shell) and the remote execution
server
# (exec) must have an entry for both the "tcp" and "tcp6"
<proto> values.
# Ftp and telnet are standard Internet services.
ftp  stream    tcp6 nowait    root /usr/sbin/in.ftpd
     in.ftpd
telnet    stream    tcp6 nowait    root
     /usr/sbin/in.telnetd     in.telnetd
Tnamed serves the obsolete IEN-116 name server protocol.
# Shell, login, exec, comsat and talk are BSD protocols.
shell     stream    tcp  nowait    root /usr/sbin/in.rshd
     in.rshd
shell     stream    tcp6 nowait    root /usr/sbin/in.rshd
     in.rshd
login     stream    tcp6 nowait    root
     /usr/sbin/in.rlogind     in.rlogind
# Must run as root (to read /etc/shadow); "-n" turns off
logging in utmp/wtmp.
uucp stream    tcp  nowait    root /usr/sbin/in.uucpd
     in.uucpd
 Tftp service is provided primarily for booting.  Most
sites run this
```

Appendix G:

```
# only on machines acting as "boot servers."
#tftp     dgram     udp6 wait root /usr/sbin/in.tftpd
     in.tftpd -s /tftpboot
# Finger, systat and netstat give out user information
which may be
# valuable to potential "system crackers."  Many sites
choose to disable
# some or all of these services to improve security.
finger   stream     tcp6 nowait    nobody
     /usr/sbin/in.fingerd     in.fingerd
systat    stream    tcp  nowait    root /usr/bin/ps
     ps -ef
netstat   stream    tcp  nowait    root /usr/bin/netstat
     netstat -f inet



# Echo, discard, daytime, and chargen are used primarily
for testing.

echo stream    tcp6 nowait    root internal
#echo     dgram     udp6 wait root internal
discard   stream    tcp6 nowait    root internal
#discard  dgram     udp6 wait root internal
#daytime  stream    tcp6 nowait    root internal
#daytime  dgram     udp6 wait root internal
chargen   stream    tcp6 nowait    root internal
#chargen  dgram     udp6 wait root internal
# The rusers service gives out user information.  Sites
concerned
# with security may choose to disable it.
rusersd/2-3    tli  rpc/datagram_v,circuit_v wait root
/usr/lib/netsvc/rusers/rpc.rusersd rpc.rusersd
#
# The spray server is used primarily for testing.
#
sprayd/1  tli  rpc/datagram_v wait root
/usr/lib/netsvc/spray/rpc.sprayd   rpc.spr
# The rwall server allows others to post messages to users
on this             machine.
 rwalld/1      tli  rpc/datagram_v wait root
/usr/lib/netsvc/rwall/rpc.rwalld   rpc.rwalld
# Rstatd is used by programs such as perfmeter.
rstatd/2-4     tli   rpc/datagram_v wait root
/usr/lib/netsvc/rstat/rpc.rstatd rpc.rstatd
```

Appendix H:

The output of the finger command

This output displays so much information to encourage a social engineering attack.

```
Login name: smith                  In real life: Sys_Adm
Directory: /export/home/smith           Shell: /bin/ksh
On since Sep  4 15:09:59 on pts/7 from 128.231.83.120
26 seconds Idle Time
No unread mail
No Plan.
```

Appendix I:

/etc/rc2.d boot scripts running on Hannibal and Sparta

Unnecessary boot scripts must be disabled to enable Apache run as a dedicated server.

```
K06mipagent
K07dmi
K16apache
K28nfs.server
K07snmpdx
README
S01MOUNTFSYS
S05RMTMPFILES
S20sysetup
S21perf
S30sysid.net
S40llc2
S47asppp
S69inet
S70uucp
S71ldap.client
S71rpc
S71sysid.sys
S72autoinstall
S72inetsvc
S72slpd
S73cachefs.daemon
S73nfs.client
S74autofs
S74syslog
S74xntpd
S75cron
S75savecore
S76nscd
S80lp
S80PRESERVE
S80spc
```

Appendix J:

Edited version of the /etc/system file without entries to ward off memory attacks.

```
*ident    "@(#)system    1.18 97/06/27 SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
forceload expects a filename which includes the directory.
Also
*    note that loading a module does not necessarily imply
that it will
*    be installed.
*    Example:
*         forceload: drv/foo
* set:
Set an integer variable in the kernel or a module to a new
value.
*    This facility should be used with caution.  See
system(4).
*    Examples:
*    To set variables in 'unix':
*         set nautopush=32
*         set maxusers=40
*    To set a variable named 'debug' in the module named
'test_module'

set test_module:debug = 0x13
```

Appendix k

Building,Installing and configuring Tcpwrappers:

Step1. Obtain TCP Wrappers source code from ftp://ftp.porcupine.org/pub/security/tcp_wrappers_<vers>.tar.gz

```
Step2.gunzip – c tcp_wrappers _<vers>.tar.gz
      tar xf tcp_wrappers_<vers>.tar
      cd  tcp_wrappers_<vers>

step3. Modify top-level Makefile
       chmod 644 Makefile ; vi Makefile
       Uncomment correct value of  REAL_DAEMON_DIR for your
       System and modify the FACILITY  variable so all
       Logging goes to LOG_AUTH
```

```
Step4. Build Software
       Add CC=gcc if gcc is the compiler.
       Make sunos5
```

Configuring TCPWrappers in /etc/inetd.conf file

# The following internet services are configured to use the TCP Wrappers daemon. Note the path to the tcpd daemon which replaced inetd as the controlling daemon for the ftp and telnet services in this example.

```
ftp stream tcp6 nowait  root  /usr/sbin/tcpd    /usr/sbin/in.ftpd
telnet stream tcp6 nowait root /usr/sbin/tcpd  /usr/sbin/in.telnetd
```

Configuring the access control files /etc/hosts.allow /etc/hosts.deny:

```
Vi hosts.allow
ftp telnet: 161.100.0.10
```

```
vi /etc/hosts.deny
ALL:ALL
```

Appendix L

Configuring a remote log host in /etc/syslog.conf

```
*.alert;kern.err;daemon.err              @sphinx
*.alert                                  @loghost2
*.emerg                                  @starhost
```

The above syslog facilities, log alert, kernel and daemon messages etc to sphinx while all alert messages and others of emergency nature are logged to loghost2 and starhost respectively. It is important to use the tab and not use the space bar in setting up this configuration.

Configuring the Network Time Protocol Network:

1).NTP Configuration of Hannibal:

```
Vi /etc/ntp.conf

restrict default  nomodify
driftfile   /etc/ntp.drift

Server      161.100.0.4 ( secondary time source )
Peer        161.100.0.2  ( Sparta )
```

2).NTP Configuration of Sparta:

```
Vi /etc/ntp.conf

restrict default  nomodify
driftfile   /etc/ntp.drift
Server      161.100.0.4 ( secondary time source )
Peer        161.100.0.6  ( Hannibal )
```

3).NTP Configuration of the internal Secondary time Server

```
Vi /etc/ntp.conf

restrict  default  nomodify

driftfile   /etc/ntp.drift
Server1     127.127.1.1 ( GPS time source in the internet )
Server2     127.127.1.1 ( stratum 5 pseudo-clock)
```

References

Practical Unix & Internet Security, 2nd Edition;
O'Reilley & Associates, Inc. April 1996

2)**Apache Web Server Administration & E-Commerce Handbook**:
Scott Hawkins prentice Hall 2001

3)**Track 6 – Common Issues and Vulnerabilities in Unix**
security Hall Pomeranz. THE SANS INSTITUTE
Tuesday, July 31 2001

4**Track 6 – Unix Security Tools. Hall pomeranz.**
THE SANS INSTITUTE. Wednesday, August 1 2001

5)**Solaris operating Environment Security**.Alex
Noordergraaf and Keith Watson. Sun BluePrints
OnLine January 2000

6)**Track 6 Running unix Applications Securely, 6.4**; Lee
Brotzman & Hal Pomerantz; THE SANS INSTITUTE, Friday
August 3 2001

7) **Track 6 Topics in Unix Security, 6.3**; John Green & Hal
Pomeranz. THE SANS INSTITUTE, Thursday August 2, 2001.

8) **Solaris Security step – By - step; VERSION 2.0** , edited by Hal Pomeranz, Deer
Run Associates THE SANS INSTITUTE 2001.