Global Information Assurance Certification Paper

Securing UNIX
GCUX Practical Assignment
Version 1.8

Security Audit of GIAC Enterprises

Lawrence van der Meer
ACME Server Security, Inc
December, 2001

Table of Contents

Executive Summary

Due to the terrorist events of September 11th, GIAC Enterprises has contracted ACME Server Security, Inc. to perform a security audit of their servers and related policies to insure that the al-Qaida terrorist network is not using their product to distribute coded messages. While there is no evidence to suggest this is taking place, the company is undertaking this audit to boost consumer and investor's confidence in the company's IT security. GIAC Enterprises specialises in the online sale and international distribution of fortune cookie sayings, thus requires a secure environment to conduct their business transactions and secure their unique product.

This document examines some of the security problems, focused around the GIAC Enterprises web server, *panther*, and its cold standby partner, *leopard*. These servers house the GIAC Enterprises' corporate website as well, as their e-commerce website. In addition to covering the servers, the document will also cover general security violations by the physical location of the equipment. Solutions are discussed and additional practices are suggested to improve security.

The security was evaluated by an onsite visit and external probes into the network. On site visits provided us with supervised access to the servers and network. The visits also gave us an opportunity to meet with a number of key IT staff members to discuss some of their security concerns. External probes into the network consisted of a variety of controlled attacks, including a vulnerability scan to determine the "obvious" problems that are easily discovered with a quick network scan.

While the overall security of GIAC Enterprises' network is quite good, there are definitely some areas for improvement. For example, there are servers are running operating systems that have not had security updates performed on them for several months, if not years. Also, some application versions are significantly out of date and some contain documented security vulnerabilities. Web servers used to conduct financial transactions are not doing so in a secure manner and can easily be upgraded to a more secure model. A number of simple changes to the environment can make a significant increase in the reliability and security of the environment at GIAC Enterprises.

Existing Infrastructure Overview

The majority of the server hardware at GIAC Enterprises are a manufactured by Sun Microsystems, ranging from an older Sun Ultra 2 Firewall to a Sun Enterprise 4500 Database Server. The web servers we are investigating for this report are a pair of Sun Enterprise 3000's – a primary server with a cold standby in case of disaster.

The web servers exist in a medium security zone between 2 firewalls. They are protected from the Internet by a solitary Cisco PIX 525 and isolated from the GIAC Enterprises internal network by a Checkpoint Firewall. A database server, used to store the fortune cookie sayings and financial transaction data, resides on the GIAC Internal Network behind the Checkpoint Firewall.

Table of Existing Hardware/Software

panther – Primary Web server (Sun Enterprise 3000)
Operating System: Solaris 2.6
Applications: Apache 1.2.6
wu-ftpd 2.6.0 (Beta 1)

leopard – Cold Standby Web server (Sun Enterprise 3000)
Identical Configuration to *panther*

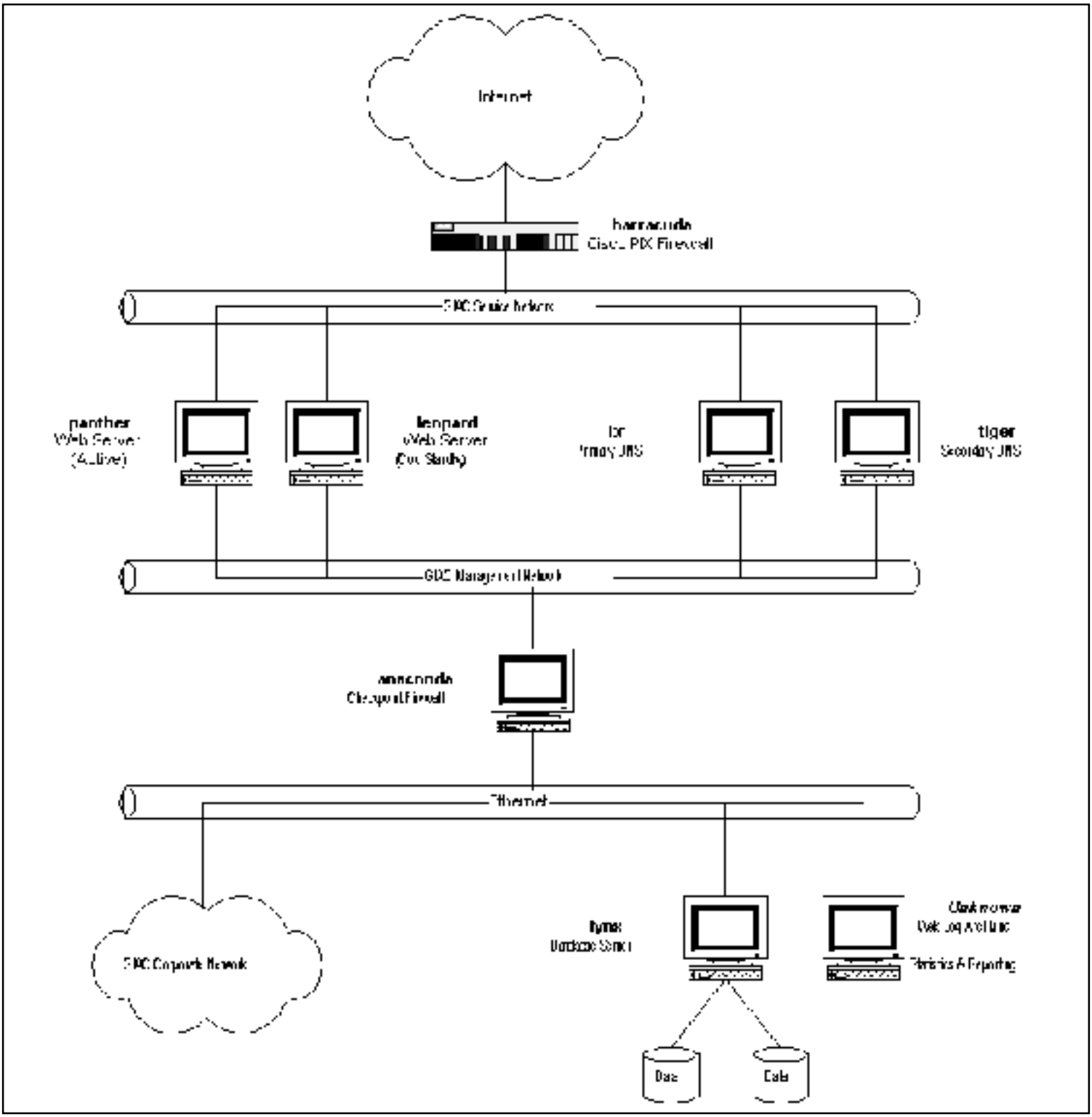
lynx – Database Server (Sun Enterprise 4500 + 2 A5000 Photon Storage Arrays)
Operating System: Solaris 2.6
Applications: Oracle 8.1.6

barracuda – Firewall (Cisco PIX 525)
Operating System: PIX Proprietary OS, Version 5.3.1 (203)

anaconda – Firewall (Sun Enterprise Ultra 2)
Operating System: Solaris 2.6
Applications: Checkpoint Firewall-1 4.1 SP4

lion – Primary DNS Server (Sun Ultra 10)
Operating System: Solaris 2.6
Applications: Bind 8.2.2

tiger – Secondary DNS Server (Sun Ultra 10)
Identical Configuration to *lion*



GIAC Enterprises Web Hosting Environment

Facilities & Disaster Recovery

Physical Security of Server/Building

Employee/Visitor Identification

On several occasions, I noticed employees without any visible identification. While interviewing staff, it was noted that it was corporate policy to require everyone to wear visible identification, but this policy is not enforced.

Comments:

- Incorrect or Absent Identification
  Corporate policy which requires all employees & visitors to require visible identification at all times should be enforced rigorously. Visitors should be required to wear special identification, to easily identify them as a guest, and should not be permitted unescorted access to restricted areas.
- Empower Employees to Challenge People with Incorrect/Absent Identification
  Employees should be encouraged to challenge people without acceptable identification. On more than one occasion, I was able to enter the equipment building unescorted and unchallenged. Employees should also have easy access to a medium to have people removed from restricted areas as necessary.

Access Points

A guard is monitoring all traffic to and from the corporate office tower, as well as the door into the equipment building. During normal business hours, the officer does not challenge a single person – but is willing to assist you in locating the party you are trying to reach. After hours, the security is much tighter as identification badges are checked before you are granted access to either the equipment building or office tower.

Crypto-keypads are present on the doors throughout the equipment building. By having keypads present at several different locations (i.e. the main door, the door to the floor and the door to the server room) corporate security is able to grant access to particular areas of the building based on role and need to a reasonably fine measure. Crypto-keypads are also present at most outward facing doors of the equipment building, allowing for multiple points of access. Most outward facing access points have surveillance cameras, which are monitored by the guard.

NOTE: While an extensive survey of the office tower was not done, the floors that were visited also had crypto-keypads, though they were not active. During interviews with staff, it was revealed that these particular keypads only become active after hours.

Comments:

- Uncontrolled Access to Office Tower
  As mentioned above, the guard does not restrict access to the Office Tower or Equipment Building during the day. The Equipment Building is protected by Crypto-Keypads, but this can be easily bypassed. (See Below.) The policy of unchallenged access should be reviewed after Mandatory Identification has been instituted.
- Over Abundant Courtesy
  Frequently I noticed that employees hold keypad controlled doors open for other employees who appear to be headed in the same direction. These same employees did not always check for proper identification from the person that they let bypass the keypad. This behaviour should be discouraged and employees should be encouraged to enter their own access code.
- One-Way Doors Installed in Equipment Building
  With keypad access provided at many of the secondary entrances to the equipment building, people are able to bypass any identification checks being performed by the guard at the main door. This coupled with the over abundant courtesy, and unauthorised people can obtain access to the building **after hours** without being challenged. All external-facing doors should have their keypad access disabled thus forcing people to use the main entrance to the building. In addition, door alarms should be installed so that the appropriate personnel can be alarmed when doors are ajar.
- Glass Doors protecting some Restricted Access Areas
  While these doors are certainly more attractive than the standard security doors, they are also much easier to bypass. (i.e. break the glass) Most of these doors do have their access points monitored by security cameras, however it was noted that there is at least one glass door that is not. If possible, these doors should be replaced with more secure doors.

Fire Suppression

Recently, GIAC Enterprises undertook a multi-million dollar project to upgrade their fire detection and suppression systems upgraded and to add a smoke evacuation system. The new system has smoke and fire detectors which are much more sensitive than the older system[1] and in many more locations, such as under raised floor tiles and in the ceiling.

The main concern with the new suppression system deployed in the equipment building is that it is a water-based pressurised sprinkler system. A better choice for a building that is primarily filled with critical electrical equipment would have been a "dry" system, such as a carbon dioxide-based system. A system such as this would not damage the systems should the system develop a leak or accidentally discharge.

Backup Power

[1] While I was present, an employee opened a bag of microwave popcorn at their desk and the steam set off the heat detector above his desk.

GIAC Enterprises' equipment building has quite a robust uninterruptable power supply (UPS) consisting of battery backup and a diesel generator. When power fails, the equipment plugged into the UPS plant switch to battery backup. If the outage lasts for more than a couple minutes, the diesel generator fires up and begins supplying power to the grid.

While I did not see this myself, employees complained about 2 things pertaining to UPS power. First, there is equipment plugged into the system that should not be. Second, there does not appear to be sufficient battery backup for the amount of equipment utilising the system. As a result of these comments, a review of the current state of backup power in the equipment building should be done, focusing on these 2 concerns.

Disaster Recovery & Backup Policies

Backups

GIAC Enterprises currently utilises 2 modes to backup their servers: Local tape drives and an off site network backup, connected via a private network connection. Local backups are performed with shell scripts using the *ufsdump* command. Full backups are done quarterly and appropriate incremental backups are being done on a weekly and daily basis. Tapes are being rotated on an annual basis and are stored in a vault for protection and security until they are needed. Remote backups are done via ADSM, with the remote server running IBM AIX on a remote IBM RS/6000. As well, the ADSM client daemon software must be installed and running on all of the servers wishing to use this service.

Comments:

- Media Disposal
  From the staff that I spoke with, they were not clear what happened to tapes that were disposed of, either from the local backups or the ADSM backups. Policy should be laid out that any backup media are destroyed before leaving the company's facilities.
- Network Backup inherently Insecure.
  Information can be intercepted as it is sent across the network while network backups being performed. This risk is minimised by the fact that the network connection is through a private network. Additional security could be employed if the data stream between the ADSM client and server was encrypted. Consider contacting your Network Backup provider to discuss this as an option if you are concerned with internal security.
- Network Backup Outsourced
  Insure that the data is being handled appropriately by the contracted company and that their systems are as secure as you would expect. With the data in the hands of a third party, consider putting strict non-disclosure and service level agreements in place, with the ability to perform independent security audits of their facilities.

Disaster Recovery Plan

GIAC Enterprises lacks a complete and up-to-date disaster recovery plan. Current documents were created to deal with any problems encountered by the Year 2000 rollover and the majority of these documents have not been updated since then. They also do not deal with other disaster

types, such as fire or flood, or any sort of hardware failure. Interviews with staff also reveal that the Y2K plans were documented, but none of them actually tested.

Current hardware disaster recovery for the *panther* web server requires the service to be restored to the cold standby server, *leopard*. The cold standby maintains a functional operating system and web server, but content would have to be restored from the last backup – either from the network backup or from local tape.

Comments:

- Validity of current Disaster Recovery Plan.
  Current documents have not been updated since late 1999. It is highly recommended that a team is immediately put together to develop and maintain the disaster recovery plan for the company.
- Testing of Disaster Recovery Plan
  Part of the Disaster Recovery Plan developed should be routine testing of the plan. This insures that everyone is aware what he or she is supposed to be doing and how to do it should a crisis situation arise. It also tests the validity of the any backups that have been done so that you don't discover that your backups are corrupted for the past 6 months while the service is completely unavailable. Accept a certain amount of risk and tear down/rebuild the *leopard* server on a quarterly basis as a 'fire drill' and make it production ready.

Vendor Support Contracts

A source of assistance that is often overlooked is the vendor of the troubled product. GIAC Enterprises, fortunately, acknowledges the importance of strong vendor-consumer relationships and maintains many high-level support contracts with the major vendors. However, it is quickly noted that there is really no way to know which servers are under what contract and whether you qualify for service or not. The Hardware/Software Inventory database contains much of this information, but it has it's own problems. (See below)

Comments:

- Employees do not understand Existing Policies
  In addition to a central repository for server contract information, the primary administrators for a server should be briefed on the coverage that their servers currently have and updated any time that the contract is modified.
- Physical Marking on Servers denoting Contract Priority
  Consider some sort of identifying mark to give people an idea as to the level of support required on a server. An idea to consider is to use red (bronze), silver and gold stars stuck to the front on the server's case to denote the class of service. This allows for an administrator to quickly assess whether they should investigate a support contract further to see if support is indeed available. (For example, if a server that has a red sticker on the front has a hardware failure during the middle of the night, they know not to bother things until the next morning because the priority is low.)
- Review Support Contracts Annually
  Frequently, the list of servers/services covered by support agreements can get out of synchronisation, so it is important to discuss your contracts with your vendors on a regular

basis. This also allows you do develop the relationship with the vendors, which can be leveraged at a later date.

- Develop Corporate Policy on "Pay-per-Incident" Support
  Have a clear-cut policy on when and how to use the "Pay-per-Incident" Vendor Support Lines and keep track of these calls. By tracking the calls, it will allow you to identify if a server or service needs to have a support contract added or increase the coverage of an existing support contract.

Hardware/Software Inventory

While on site, I was able to examine the Hardware/Software library for GIAC Enterprises. Currently they have their data in an Access Database on a single PC. While there is a lot of information in the database, the employees I spoke with complained about the fact that the data is out of date and hard to get at. By putting relevant information at the fingertips of individuals, you will enable them to answer support questions (both from internal customers and the vendor alike) in a more expedient manner.

Comments:

- Migrate Database to a more centralised location.
  What good is this information if the employees cannot draw upon it? Consider moving the database to a more accessible location – like an Oracle Database (Software which is already deployed within the company).
- Update & Maintain Inventory Database.
  To make the data useful, it must be maintained on a regular basis. Once the database is much more accessible (see above), individual systems administrators can keep their data current through a similar interface.
- Insure Easy Access to View and Update Data
  There's no point in developing a centralised database if you're not going to make it accessible to employees who need the data. To make this easily accessible, consider deploying a web-based intranet for accessing the data. By developing and maintaining the access to this data, you will be able to control what level of security and access you provide to who.

Staff

Most of the staff that I interviewed at GIAC Enterprises was content in their current position. Management clearly communicates what is expected of their staff and maintains an 'open door' policy. This allows the lower level managers to have a very accurate perception of the current attitude of most staff members.

Some minor suggestions that could improve security:

- Security Background Checks/Bonding
  This would give you a good understanding as to whether a candidate or employee should be

considered a security risk. If the company does not wish to deploy this level of scrutiny corporately, they should at least consider the employees working with sensitive material, such as the corporate firewalls or the database servers containing financial data.

- Skill Level/Training Requirements
  Frequently, the first thing to get cut as part of corporate budget reduction is the training budgets of most departments. *I cannot stress how important it is for the IT staff to maintain their skill levels with adequate training.* Be sure to supplement any personal development with professional training – using a whet stone will keep a knife sharp, but once and a while it's nice to see a professional to put a good edge back.
  If adequate training is not available, be sure that you are hiring staff who have the requisite skills to do the job to the appropriate level.
- Adequate Staff Levels.
  There is no such thing as a 'one man army' in the IT field. Day to day operations of a system is much work, and sometime more, than the initial design and deployment of the server. Be sure that the staff you have are not overworked as even superior staff members can get sloppy when under sufficient stress. And sloppy work can frequently as bad as having an inexperienced administrator on the job – though an inexperienced administrator will improve with time. A superior admin under stress with degrade even further.
- "On the Job"
  Be sure that you are making your staff members fully aware of all the rules and regulations (corporately and legally) around the servers they are responsible for. Never assume that someone knows what those rules are. Have privacy & regulation sheets circulated and signed annually to make sure that not only are people aware of these regulations, but that they are continuously reminded of them as well. Have the employee sign a declaration that they have read and understood these rules and keep these on file for future reference.

Administrative Practices & Current Standards

Not everyone considers the impact of something that they do on a day-to-day basis. Even worse are the things that are *not* done on a day-to-day basis, things that probably should be done in order to maintain a minimal level of confidence in your level of security.

One of the most common excuses encountered by IT Professionals questioning a particular practices is that "this is the way it was always done". Experienced Administrators should be encouraged to speak up when they see a particular standard or practice that could place the corporation at risk.

In this section, we outline some of these practices and standards that we discovered through company documentation, administrator interviews and exploration of the servers.

Host Naming Convention

One of the easiest defence tools to employ is often the most overlooked – that tool is deception. When used in combination with other security tactics, it is a valuable tool. In naming your servers, set-up a hostname convention that does not indicate the role of the service.

Reviewing the GIAC Industry Network, it is obvious to see that the names of the servers (for example, panther and leopard) have no bearing on their role as web servers. Additional network components, such as the firewalls and database servers, also have similar non-descriptive names.

Comments:

- Develop a Documented Policy on Host Naming Conventions
  Bring the appropriate members of the company together and come up with a corporate standard on host naming that will balance security with ease of use. Publish a policy that will be used as part of server configuration standards.
- Consider Randomizing Server Names
  Right now, servers of a similar nature are named under the same naming convention. All firewalls are names after snakes and all application servers associated with the sale of online fortune cookies are names after wild cats. Consider randomizing your hostnames such that naming conventions cannot be used to associate servers together for similar functions or part of the same process (as they currently are).
- Consider "Coded" Hostnames
  While real names make things easy to remember, it can become impractical for controlling the inventory of a large quantity of servers. To deal with this consider using a 'code' to develop the name for your server. By using a coded name for the server, you can further increase the level of deception of the server. For example, if you had a server called *ssh-sjnb00*, it is quite difficult to know what this server is. (The simple code shown identifies the first **S**un **S**olaris **H**TTP (Web) Server located in **S**aint **J**ohn, **N**ew **B**runswick.) Also, as the above example illustrates, you can also embed significant amount of information into a coded

hostname that can actually aid in inventory control.

NOTE: Should the code used for your naming convention be broken, you could have inadvertently provided a hacker with additional information to target their attack. Consider using this method on internal networks/servers only.

Operating System and Application Versions

During my visit to GIAC Enterprises, we discovered that current corporate standard that require Solaris 2.6 be deployed corporate-wide. In discussions with administrators, and reviews of corporate documents, we are told that the systems panther and leopard conform to the standard. We verified this with command *uname –a*, which provided us with the following information:

```
SunOS panther 5.6 Generic_105181-19 sun4u sparc SUNW,Ultra-Enterprise
```

The format of the command's output is roughly:

[OS Name] [Node Name] [OS Release] [Kernel Version] [Kernel Architecture] [CPU Type] [Hardware Platform]

This particular version of the Solaris Operating System was released in the mid 1990's and most of the corporate documentation reviewed had not been updated since that time as well. The most recent version, Solaris 2.8 (marketed as Solaris 8), was released in late 1999 and would be a better choice as corporate standard.

In addition to the operating system being out of date, the major applications that were audited were also not running the most current release versions. *Leopard* and *panther* were both running Apache 1.3.4 and wu-ftpd 2.6.0 (Beta 1) – both of which are several versions behind the most current.

The versions were identified by executing the following commands –

Apache

```
# ./httpd -v
Server version: Apache/1.3.4 (Unix)
Server built:   Mar  2 1999 21:23:25
```

wu-ftpd

```
# ftp panther.giac-enterprises.com
Connected to panther.giac-enterprises.com.
220-########################################################################
220-
220-This system is restricted to authorized personnel.  Only
220-authorised work is to be done.  Use of this system is an agreement
220-to monitoring.  Any unauthorized use is subject to disciplinary action.
220-
220-Access Granted for IP Censored.
220-
220-########################################################################
220 panther.giac-enterprises.com FTP server (Version 2.6.0(1) Tue Nov 30 15:12:54 AST 1999) ready.
Name (panther.giac-enterprises.com:lavander):
```

As of this report, the current stable release of the Apache web server is 1.3.22 and wu-ftp is 2.6.2. In addition, there has been a number of security vulnerabilities reported for wu-ftp since

that date. The wu-ftp website has the CERT Notification and patches to close the vulnerabilities available from their website – http://www.wuftpd.org.

Comments:

- Examine Possibility of upgrading Operating System/Applications to a more recent version
  Being on the cutting edge of technology is not a good idea, but it is always a good idea to stay close to current on most things. Solaris version 2.8 (marketed as Solaris 8) is almost 2 years old now and should be "mature" enough to be considered for most production environments. Upgrading to more recent versions of Apache and wu-ftp should also be considered, perhaps at the same time.
- Establish Standard "Upgrade Policy" to keep OS/Applications at more Current Versions
  To insure that you do not fall far behind in your applications and operating system versions, establish a standard policy about when and how to upgrade versions. This document will be similar to the 'Regular Patch Schedule' (as patching is essentially a minor version upgrade), but will be employed on an infrequent basis. I would suggest that all major application versions be audited quarterly and appropriate action (as outlined in the upgrade policy) taken at that point.

Operating System Patching

Administrators disclosed that most systems have not has Operating System patches installed since the company's Year 2000 Readiness Plan formally wrapped up early in the new millennium. This statement is confirmed by looking at the last time *panther* was last patched - March 28th, 2000. This was determined by looking at the last modification date of the patch database file: /var/sadm/patch/*.patchDB*. (This file contains the information on which patches have been installed on the system.) This is a most disturbing trend as operating system vendors try to close exploited security holes as fast as they are discovered and provide general bug fixes as well. As of the writing of this report, the Solaris 2.6 Recommended and Security Patch Cluster is approx. 60 MB, which covers a significant number of updates since the last update.

Sun has provided us with a couple ways of determining which patches are missing from the current operating system. The two most useful are a couple of tools that are available for download from the SunSolve Online web site (http://sunsolve.sun.com). First is *patchdiag* – created by Sun to quickly analyze your system against a list of known patches (patchdiag.xref) and determine which patches are missing, obsolete or out of date. The resulting list can quickly determine how badly your system is away from current versions. A newer tool called *patchcheck* has now replaced patchdiag as Sun's recommended patch analysis tool. Patchcheck has added the ability to create a HTML marked up document to easily download the required patches after running the script. For the purposes of a quick examination of the state of the server, we continue to use patchdiag tool since we do not plan on downloading any of the suggested patches.

The output of patchdiag from *panther* can be found in the appendix. A quick glance through the resulting list will show you a number of problems – for example, current revision of the kernel is 11 revisions behind and there are a number of buffer overflow security patches missing from the system all together.

Comments:

- Immediately apply Operating System Patches to Servers
  In the 2 years it has been since the last patch update, there have been a number of major security exploits discovered in some of the standard operating system tools. Patches have been released to close these exploits and should be applied immediately. The *Recommended and Security Patch Cluster* for Solaris 2.6 can be downloaded from the Sunsolve Website. After the Patch Cluster has been applied, you can run *patchcheck* to determine if there are still patches required that were missed by the cluster. Using the HTML-marked up output that is generated by this command, along with your SunSolve Online username and password (provided with your support contract) - will allow you to easily download the missing patches with ease.
- Establish Regular Patching/Update Schedule
  Work together with all required players to determine a regular patching schedule to maintain the patch levels of your server. One of the first questions you are typically asked by vendor support staff is what your system patch level is. By establishing a regular schedule, you are likely to be quite close to the most recent patch levels. Included in this plan would be a notification process (so that everyone is aware that the patching is happening) and a general implementation/backout plan – with any 'special' procedures to be documented closer to the day of the patch. I would recommend a schedule of monthly updates to insure that your systems are kept current.
- Establish "Rapid Reaction" plan to deal with Major Security Vulnerability Notifications
  Similar to the "Regular Patching Schedule" described above, develop a plan that will allow you to apply "emergency" patches at a moments notice to deal with major vulnerabilities.
- Monitor Appropriate Mailing Lists and Websites for New Security Vulnerabilities
  Encourage administrators to monitor major mailing lists and websites for new security vulnerabilities. If existing workload does not permit this, designate several administrators to watch these lists/sites for the corporation and make it their responsibility to inform the appropriate people if they receive a notice that concerns them.

Software Build and Version Control

With regular upgrades scheduled (or the policy being developed), the company will need a plan to deal with building and controlling the new versions across multiple servers. Currently, the individual administrators install a binary, if possible, on the server and install/configure it from there. If no binary is available, they download the source code, appropriate compiler and build it on the server, then install and configure the application.
In order to insure that all corporate servers are running the same software version, a software repository has to be set up. This will be an internal location for all administrators to obtain (and build, if necessary) their software. A team of senior administrators should be selected to maintain the repository, building the binary distribution from source code as required. Other administrators access the repository via FTP and download the required binaries from there. If a required application or version is not present in the software repository, a request is sent to the repository administration team to have the application added.

Comments:

- Establish a Software Repository
  If possible, obtain a dedicated server with ample disk space to house the software repository within the internal corporate network. Staff the repository administration team with senior administrators and come up with a policy for adding software to the repository. Administrators should be using this repository to access all software to be installed on their servers rather than versions downloaded directly from the internet.
- Establish Corporate Policy on running Pre-Release Software
  It was noted that the version of wu-ftp that was installed on the 2 web servers, *leopard* and *panther*, was pre-release (i.e. beta) software. It is recommended that the corporation make a written policy about the use of pre-release software in the production environment. It is strongly encouraged that, for the sake of stability, that only "stable" release versions be used.

Access Rules & User Limitations

Access control is one of the most important things one can tighten down in a system. During my visit to GIAC Enterprises, I noticed that while a number of areas are already controlled by corporate policy, there are still a number of areas for improvement. GIAC's current access control policy has a number of restrictions, including the following:

- No Root password access beyond the Server Administrators
- Network Access to Servers is controlled by Firewalls, Network Appliances (routers, et al.) and some servers deploy TCP Wrappers
- Access to corporate network, from outside the physical facilities, are limited to dedicated dialup (using a SecureID One-Time-Password System) or Virtual Private Network (VPN) services.

The 2 web servers that are under investigation are currently running TCP Wrappers to control access to their telnet daemons, but leave wu-ftp unprotected. In conjunction with some static routes on the server, TCP Wrappers insures that telnet sessions are only accepted from the management interface of the servers.

Through interviews with staff, it was determined that there currently is no corporate policy set on requirements for passwords – superuser or normal user alike. While root passwords were quite patterned, several administrators admitted to me that they use common words or acronyms for ease of use as their own passwords. Others mentioned that they rotate their passwords between 2 or 3 passwords on a monthly basis, thus repeating the same password every 2 to 3 months.

Comments:

- Version of TCP Wrappers deployed extremely old
  Again, the last time that TCP Wrappers had been upgraded on these servers was to accommodate concerns about Year 2000 compatibility, and thus the version deployed is several years old. Administrators should consider upgrading to a more recent version of the software.
- Deploy Host-Filtering Software across Corporate Network
  Most servers that are behind a firewall are not running server-based host filtering software

because the Server Administrators and Network Designers believe that this is adequate protection. By deploying host-filtering software with a restrictive access policy, you can insure that servers are protected even if the corporate firewalls are breached. It is highly recommended some sort of host filtering software, like the freely available TCP Wrappers, be deployed throughout the corporate network.

- Audit Host-Filtering Access Lists on a Regular Basis
  In large data centres, like that of GIAC Enterprises, it's amazing to see the turn around on deployed servers. As a result, 6 months down the road, the server you may have granted access to through your host-filtering software may no longer exist within the data centre. By performing regular audits of the access control lists, you can insure that there is no inappropriate access being granted to your server.
- Consider deploying SUDO
  A useful tool to consider deploying is SUDO. This allows an administrator of a system to grant non-root users the ability to run some (or all) commands as the super-user. This is a very powerful tool, but one where you can easily shoot yourself in the foot. Interviews with administrators indicate that to date, there has been no requirement for the deployment of this tool. One scheme for deployment is to use it to grant super-user privileges to all administrators such that they can execute all necessary commands through sudo. This way, the root password is only used in times of emergency.
- Consider running Password Cracking Programs
  By randomly running a password cracking program, like John the Ripper, on servers' password files, you will be able to insure that all users are using passwords that are difficult to guess. Consider running these on a regular schedule, against random servers in the environment such that administrators will not know when the scan will take place.
- Consider expanding One-Time-Password System.
  As mentioned previously, the corporation already has a SecureID system deployed for One-Time-Password access to their corporate network. A one-time-password system generates random sequences of numbers that are synchronized between token cards (the SecureID card) and the server. These random numbers are appended to a personal identification code to form a person's password. A hacker would have to guess someone's identification code as well as steal that person's token card to know his or her password. Sniffing the person's unencrypted telnet session would get a hacker the identification code, but the random number from the card would likely have changed by the time the hacker had a chance to use the password. The company should consider deploying this functionality throughout the servers in their data centre for additional protection of their servers.
- Corporate Policy on Passwords
  The appropriate people within the organisation should create a documented policy on password requirements. Topics included in this policy would be things like:
  - what constitutes a "good password"
    (i.e. inclusion of non-alpha characters; not a dictionary word)
  - minimum number of characters required in a password
  - minimum amount of time allowed between password changes
  - maximum amount of time allowed between password changes

  If possible, this policy should be included in the corporate information technology security policy – which every employee should be required to sign off on an annual basis.

Insecure Network Communications

The majority of services using TCP/IP are insecure. A telnet session, for example, watched by a packet sniffer (and thus, any passwords being sent through it are exposed) and certain implementations will also allow for the session to be hijacked through use of predictable sequence numbers. To that end, you have to be sensitive of the data being sent across the network and the method in which it is being sent.

GIAC Enterprises is currently using insecure network communications for most of their services, notably telnet and http (web) traffic and should be upgraded to more secure options

Comments:

- Consider Installing Secure Shell to replace traditional Telnet Daemon.
  By deploying Secure Shell (SSH), you create an encrypted tunnel between the client and server such that the data stream can no longer be viewed with a traditional packet sniffer and your sessions are now much more difficult (if not impossible) to hijack.
- Consider Installing Secure HTTP to protect Financial Transactions
  By deploying web server software capable of Secure Socket Layer (SSL) transactions, you can greatly increase the security of all transactions between the client (web browser) and the web server. The SSL layer is an encrypted tunnel, much like with SSH, through which data flows between the client and the server. Thus where financial transactions are taking place, you can secure the data (like credit card information) submitted by your clients as it travels to your web server.
  NOTE: This only encrypts communication between the client and web server. Any further communications – for example, between the web server and database server – would have to have it's own communication encrypted.

General and Common Gateway Interface (CGI) Script Security

Since GIAC Enterprises has a strong online presence with their fortune cookie business booming, a logical target for any attack is to exploit something that has been deployed within their website – and the most commonly exploited "tool" here are mis-configured CGI directories or insecurely written scripts.

GIAC Enterprises has some talented programmers on contract and on staff to maintain their e-commerce website and I was pleased to see that their CGI applications have been created with the utmost care in mind. The majority of these CGI applications have been written in C, thus reside on the server as binary code, with a few other minor and 'helper' scripts written in Perl and Borne Shell. All Perl scripts were run with a number of options. First, they were run with taint checking enabled. This encourages the programmer to test any input being read into the script from the 'outside' to check to see if it is valid. Second, they enabled 'strict' mode, which makes Perl behave much more 'C' like, requiring all variables and the like to be declared first. I did not have access to the C code for the various C applications (the company said it was outside the scope of my audit), however, from what I saw of the Perl code there is little to worry here.

Similar to CGI Scripts, I looked through a number of administrative scripts, like the backup cron script, and many of these were too simple to exploit. That being said, someone might throw together a 'simple script' some day that opens a major security hole that the administrator in question is unaware of.

Comments:

- Develop Checklist for Known Security Holes
  Have qualified personnel develop a 'check list' of easily identifiable security holes from the application code. Consider the WWW Security FAQ (URL listed in appendix) as the foundation for this checklist. This will allow staff to quickly identify problems with the code that can then be documented and returned to the source for correction.
- Audit All Applications (CGI/General) before Deploying.
  Based on the Checklist developed above, administrators will be able to quickly audit all scripts/applications before they are deployed into production. All applications accepted into the production environment should be documented as being audited and the auditing administrator identified.

Added Security through Better Initial Configuration

Due to the limited security measures currently deployed within the environment, we made the assumption – as we do with all of our audits – that the system has already been compromised and base our recommendations from that point. Systems that were examined were done using a CD of 'known good commands' to insure that the binaries being executed were not trojan versions of common operating system commands – like *cd*, *ls*, etc. While we prefer to run commands from the CD, we do encounter systems where this is not possible. In these cases, we have also mounted this CD via NFS (if available) or even copied the binaries from CD to the system we are auditing. Both of these methods are undesirable since there are security problems with running NFS and the binaries could be modified by the method used to transfer the binaries from the CD to the system.

Fortunately, both panther and leopard have CD-ROM drives, thus we were able to execute the commands directly from the CD.

Although our investigations did not turn up any indication that the systems may have been compromised, we strongly suggest that these systems be built up according to a stronger security standards encouraged through this document. A good document to begin security discussions on is The SANS Institute: Solaris Security Step by Step V2.0 (edited by Hal Pomeranz), which is available through SANS at http://www.sansstore.org. This documents walks through the steps required to build a fairly secure system. The majority of the recommendations put forward in this report have their roots within this document and forms the baseline for a good security standard.

Although a complete rebuild is not totally necessary to implement the majority of the suggestions put forward within this section of the document, it is certainly a good idea so that you are 100%

sure that the server has not been compromised or poorly configured in the past. Given the fact that the hot standby server (leopard) is available, this rebuild could be effected without impact to service. The rebuild would also provide an opportune time to upgrade the operating system and major applications as suggested earlier.

Sun OpenBoot PROM Configuration Options

During the evaluation of the servers, I discovered that the server's OpenBoot PROM (also known as the eeprom or "ok prompt") was in the default security configuration. There are a number of configuration options available to an administrator to secure the PROM from its default configuration.

First, the PROM can be configured in 3 security levels: *full, command* and disabled.

Full security mode prevents access to the OpenBoot PROM and the system from being booted until a password is provided. *Command* security mode prevents access to the OpenBoot PROM until the password is provided. *Disabled* security allows for access to the Boot PROM and the system will boot normally.

The security of the OpenBoot PROM can be configured from both a root command prompt or from the OpenBoot PROM itself.

From the OpenBoot PROM:
setenv security-mode *level* (example: setenv security-mode command)
From within the Operating System:
eeprom security-mode=*level* (example: eeprom security-mode=full)

Changing the OpenBoot PROM password is done similarly:
setenv security-password in OpenBoot
eeprom security-password= from within the Operating System.

Another useful command allows you to monitor the number of incorrect attempts at the PROM Password. By typing *eeprom security-#badlogins* you can determine the number of bad login attempts have been made.

A note of caution: OpenBoot Security levels can easily be used against you should the system become compromised. Once a hacker has obtained root access to the server, they can use the above eeprom commands to set the OpenBoot PROM security to full, set the password and reboot the server. In this case, the administrator is stuck at a password prompt where he does not know the password. The only supported recovery from this is to contact Sun and order a replacement PROM.

Comments:

- Enable *command* level PROM security.
  Following the theory that a little security is better than none at all, we recommend that you

set *command* level security from the PROM. *Full* security is not convenient for a highly available server as there are times where the server will need to be booted remotely and by setting the PROM level to f*ull*, you will be unable to complete the boot sequence without having someone at console to enter the password.

- Establish Corporate Standard for OpenBoot PROM Passwords.
  It's important to understand that this password is stored in a plain text file on the system. As a result, it will not provide you with security should the server's root user be compromised. As a result, it is recommended that you establish a corporate standard for OpenBoot PROM password, such as a pattern that can be used enterprise wide. This should provide you with a password that will be difficult to guess, but "easy" to remember should you need access to the OpenBoot PROM.

Operating System Configuration

In addition to having your system patched to the most current levels (see Operating System Patching in **Administrative Practices & Current Standards**), there are a number of files that should be created or modified to enhance the security on the server. In this section, we summarize some of the more important files and what can be done to tighten security.

/etc

passwd and shadow

These two files contain the user account information (/etc/passwd) and the password/expiry information (/etc/shadow). On initial configuration the standard user accounts, such as the print daemon user (lp) and the UNIX-to-UNIX Copy Administration users (uucp & nuucp), should be evaluated for validity. Since we are configuring a server for a particular role, it is likely that some or all of these users could safely be removed from the system.

Passwords are covered in more detail in the Access Rules & User Limitations section of **Administrative Practices & Current Standards** as well as later on in our discussion on the password-cracking program, *John the Ripper*.

notrouter

The presence of the /etc/notrouter file instructs Solaris not to behave as a router - disabling IP forwarding between network interfaces. It is important for the servers at GIAC Enterprises to be configured in this manner so that packets will not be routed between public and private interfaces. If this file was missing it could be possible for an attacker to find (or create) a path into the restricted network.

ftpusers

Users included into the /etc/ftpusers file will not be able to access to the server through ftp. The root account should be included in this file, along with any "pseudo" accounts (like uucp, nobody, sys, bin and others) that are not removed by the initial audit. On *panther*, two main groups use ftp. Web developers move content into place via ftp and system administrators use it for administrative purposes. Any other accounts should be placed into the /etc/ftpusers file.

Access to ftp is further restricted via TCP Wrappers (discussed below).

inetd.conf

The file /etc/inetd.conf (a symbolic link to /etc/inet/inetd.conf) contains all of the daemons controlled by the master internet daemon – inetd. Most of the services that are started by inetd can be disabled (by adding a # to the beginning of lines corresponding to the services you wish to disable) or removed (by deleting the line entirely). For example fingerd can be stopped as shown in the following example:

```
#finger stream  tcp     nowait  nobody  /local/apps/tcpd7.6/tcpd        in.fingerd
```

After any change to inetd.conf, you need to send a "hang-up" to the inet daemon.

We strongly recommend that any services not being used be removed from inetd.conf completely and those that remain be protected via TCP Wrappers.

/etc/cron.d

at.allow / at.deny

These files control access to the *at* and *batch* scheduling facilities which allow you to schedule commands for execution at a later date. Usually these facilities are used for "one time" executions. The presence of *at.allow* denies access to everyone who is not in this file. The only time *at.deny* is consulted is if there is no *at.allow* file – in this case allowing everyone who is not in the *at.deny* file. If neither file exists, only root is allowed to execute commands with *at* and *batch*. If at.deny is present, but empty is allows global access to these facilities.

The recommended configuration would be to allow access to these facilities as the needs present themselves.

cron.allow / cron.deny

Similar to the at.allow / at.deny files, these control access to the cron facility which allows you to schedule commands on a particular interval. The allow and deny files work similarly to that of the *at* files, with the exception that there is no way to enable global access and that one of the two files has to be present for it to apply to root. Otherwise, root will always have access to cron.

Again, the recommended configuration would be to allow access to cron as the need present themselves.

/etc/default

inetinit

In stock configuration, Solaris uses a TCP sequence number that is predictable. From /etc/inetinit, description of this mode is *"Old-fashioned sequential initial sequence number generation."* If a hacker can predict the TCP sequence numbers, it is possible that they could hijack users' sessions to gain access to the server. By configuring the TCP_STRONG_ISS variable in /etc/inetinit to use mode "2" (described as *"RFC 1948 sequence number generation, unique-per-connection-ID."*) it makes it significantly more difficult for attackers to hijack sessions.

Telnet sessions are further protected by the use of SSH.

login

There are a few variables that we need to pay attention to in this file – and they are shown before in their desired configuration. Some of these are the Operating System's default settings, but they should be verified.

CONSOLE=/dev/console

This variable only permits root logons from console.
Users requiring root access must login first and then su to root.

PASSREQ=YES

This variable requires all users to have passwords to login.

TIMEOUT=300

This variable allows you to set an automatic timeout of idle connections in seconds. We recommend a shorter timeout than the default of 5 minutes, but adjust to fit your needs. Simply defining this variable to some length is a benefit.

UMASK=027

This variable sets the default permissions on any file/directory created. A umask of 027 will create directories with permissions of 750 (drwxr-x---) and files with permissions of 640 (-rw-r----). This is the recommended umask, over the default of 022.

SYSLOG=YES

This variable allows the logging of multiple failed login attempts and all root logons using the syslog facilities. The comments within /etc/default/login provide more information. We recommend that this logging be enabled for auditing purposes.

su

The only variable that you really need be concerned with here, from a security point of view, is the SYSLOG=YES line. Like in /etc/default/login, this allows for logging of most su events via the syslog facility. Comments within /etc/default/su describe the logging levels associated with the various su events. We recommend that this logging be enabled for auditing purposes.

passwd

The /etc/default/passwd file allows you to define some simple rules around passwords, such as a minimum length and frequency of required change. Variables are shown here with their installation configurations:

MAXWEEKS=

Maximum number of days that a password is valid; default is undefined.

MINWEEKS=

Minimum number of days between password changes; default is undefined.

PASSLENGTH=6

Minimum number of characters required for password.

If you plan on using password aging, you should also define:
WARNWEEKS=
Number of days that a user should be warned that password would expire.

These variables should be configured with default values that express the corporate standard that we recommend be developed in the Access Rules & User Limitations section of **Administrative Practices & Current Standards**.

cron
Insure that cron logging has been enabled by checking /etc/default/cron for the line *CRONLOG=YES*. Cron will then log in /var/cron/log.

ftpd / telnetd
These two files control the banner output and some simple configuration.
In telnet, your "banner" is located here:

```
BANNER

login:
```

For example:

```
SunOS 5.6

login:
```

In ftp, the banner is located here:

```
220 hostname FTP server (BANNER) ready.
Name (IP:user):
```

For example:

```
220 panther FTP server (SunOS 5.6) ready.
Name (172.16.0.129:lavander):
```

As you can see, these commands are offering up the version of operating system we are running. By defining BANNER in /etc/ftpd and /etc/telnetd you can eliminate this information. Typically, we define BANNER="" in /etc/telnetd to remove this banner completely and rely on the banners produced by TCP Wrappers. In /etc/ftpd, because it would look silly to have "()" in the information line, we usually define BANNER="Restricted".

Another variable that can be set in /etc/ftpd is UMASK. UMASK allows you to specify the default permissions on any file/directory created through ftpd. While this will be overridden by wu-ftp (discussed later), it's a good idea to define it here anyway. We suggest a UMASK=0027 (file permissions of 640 and directory permissions 750).

/etc/init.d/netconfig

This is not a file created by the operating system install process and typically does not exist in audited servers. This file is used to house a number of network driver tuning commands - *ndd* commands – that will secure your network interfaces from behaving in an unexpected manner.

ndd –set /dev/arp arp_cleanup_interval 60000
Shortening the arp cache timeouts can prevent arp attacks and help defend against arp spoofing (where one machine tries to assume the identity of another). The timeout for unsolicited arp information is reduced to 1 minute, from the default of 5 minutes, by setting the ndd variable *arp_cleanup_interval* to 60000.

ndd –set /dev/ip ip_forwarding 0
While we have addressed IP forwarding by using the /etc/*notrouter* file, it can also be disabled by using another *ndd* command. To insure that this server is not behaving as a router, we recommend adding this line into the netconfig script.

ndd –set /dev/ip_strict_dst_multihoming 1
A multi-homed server is one which has connections to 2 networks. In the case of GIAC Enterprises, they use a service (public) network for customer access to their servers and a management (private) network for maintenance and administration. This *ndd* command will prevent the systems from accepting packets on one interface that appear destined for the other. Even with IP forwarding disabled (above), multi-homed Sun servers will , by default, forward packets to other hosts on the local network – which could potentially bridge public and private networks.

ndd –set /dev/ip ip_respond_to_timestamp_broadcast 0
A server usually is not required to respond to timestamp requests broadcast to the network. Responding to these requests could lead to DOS attacks using these broadcasts to tie up network resources. Unless you have a need for this functionality, we strongly recommend that you disable this functionality using the *ndd* command above.

ndd –set /dev/ip ip_respond_to_timestamp 0
Individual timestamp requests could be a normal part of network traffic, however if single timestamp requests are not needed then the above *ndd* command should be used to disable them.

ndd –set /dev/ip ip_ignore_redirect 1
This *ndd* command causes the server to ignore ICMP redirect packets that are usually sent from a router to specify an alternate path. Unfortunately, attackers can spoof these packets to cause the server to redirect traffic out a different gateway as either a Denial of Service attack or possibly to intercept the data. This setting is strongly recommended.

ndd –set /dev/ip ip_send_redirects 0
Since only routers should be sending these ICMP redirect packets, we strongly suggest disabling this functionality so that hackers cannot use your server as a platform to launch this sort of attack..

The /etc/init.d/netconfig script should be run after /etc/init.d/inetinit, which usually runs as /etc/rc2.d/S69inet. We recommend linking this file into /etc/rc2.d as S69netconfig using the command *ln /etc/init.d/netconfig /etc/S69netconfig* after /etc/init.d/netconfig has been created.

/home/*user*/.rhosts

The .rhosts file allows users to grant permission for users on remote hosts to connect to the hosting server without providing a password. (Users/systems defined in a .rhosts file are considered "trusted".) Plus signs (+) are used as wildcards in a .rhosts file, so an entry of "+ +" would let any user from any machine log in without authentication. Since we know that we should not trust anyone, we want to discourage users from using this style authentication.

First, we recommend editing /etc/pam.conf to remove .rhost-style authentication. The easiest way to perform this task is to use the following commands:

Backup pam.conf	cp –p /etc/pam.conf /etc/pam.conf.*date*
Insure that it's not "easily" restored.	chown root:other pam.conf.*date*
	chmod 0000 pam.conf.*date*
Remove rhosts_auth from pam.conf	grep –v rhosts_auth /etc/pam.conf > /etc/pam.new
Install new conf file	mv /etc/pam.new /etc/pam.conf
Insure correct owner/permissions	chown root:sys pam.conf
	chmod 644 /etc/pam.conf

Since we also recommend upgrading to SSH, it is worth mentioning that SSH does not use the settings from pam.conf. The file is only edited in case the system ever is reverted back to using the r-commands (rsh, rlogin, etc) instead of their more secure SSH counterparts.

Next, we want to track down any rhost files currently on the system. Since these files must reside in the user's home directory, we can use a simple find command to track most of them down.

```
find /home -type f -name .rhosts
```

Since these files can also be used with the root user, a manual inspection of the root user's home directory should also be performed.

Fortunately, there were no .rhosts files on the servers when we checked. In order to keep users from making them we suggest creating empty .rhost files, which are owned by root, in each user's home directory. This can be accomplished by the following quick set of commands:

Create or replace user's .rhosts file with null	cp /dev/null /home/*user*/.rhosts
Change ownership/permissions	chown root:root /home/*user*/.rhosts
	chmod 0000 /home/*user*/.rhosts

Repeat for all users, including root, on the system.

Of course, this will not stop anyone that has root access, including attackers that come by the privilege, from modifying these files. We hope that legitimate users will be discouraged from using the .rhosts file – and those who persist can be dealt with under corporate policy dictating these files cannot exist. As to the hackers who compromise the server, we hope that this change will slow them up for an internal process to be alerted to the change and the break-in detected before too much damage is done.

Another file similar in function to the .rhosts file is /etc/hosts.equiv. The major difference is that this file allows global access from the machines listed without

prompting for a password. This file was also not detected on the servers, and an empty file also was created in /etc (using the same procedure as with the .rhosts files) to prevent casual creation of the hosts.equiv file.

/var/adm/loginlog

By creating this file, it enables logging of failed logons. This file does not exist by default, and did not exist on either *panther* or *leopard*. We created the files on these servers at that point using the following commands:

Create the empty loginlog file	touch /var/adm/loginlog
Change ownership/permissions	chown root:other /var/adm/loginlog
	chmod 600 /var/adm/loginlog

Permissions are set to read/write for root only, since there is no need for "regular" users to have access to this file.

Running Unnecessary Services

In order to maintain a high level of security, systems should be configured as 'role based' servers. This means insuring that no unnecessary services are running on the server. System documentation, as well as the output from the *pkginfo* command, shows that the GIAC web servers have been installed with more than the "minimum" Operating System. In fact, the two web servers examined showed little indication that they had been configured as dedicated web servers as many default services were running at the time of the audit.

Comments:

- Disable any services that are not necessary for the Role of the Server.
  By configuring a server such that it has the minimum number of services to do its job properly, you are reducing the number of applications that a hacker can try to exploit. This is usually done by identifying the associated start/kill scripts within the rc directories (/etc/rc[0-3,S].d) and "x"ing them out.

  Files within these rc directories are executed while changing into that particular run level – for example, on a default install, the boot process has the system run through all the start scripts (anything starting with an S) in rc2.d and rc3.d. (This behaviour is defined in */etc/inittab*.) On leaving a run level, the kill scripts (anything starting with a K) are run. Both start and kill scripts are run in the order they are listed in – thus (for example) S75cron would be run before S88sendmail. Any file that starts with anything other than an S or K are ignored. To disable a service, we recommend simply renaming them to the same filename with a preceding "x". Thus, S88sendmail would become xS88sendmail.

  To disable a service, follow these steps. We will use "sendmail" as an example:

Manually Stop the Service	/etc/init.d/sendmail stop
Change permissions/ownership to discourage use	chown root:root /etc/init.d/sendmail
	chmod 0000 /etc/init.d/sendmail
Find all references to the init script and X out –RC Scripts are (usually) hard links	
Identify the i-node number	ls –i /etc/init.d/sendmail
Find all files with the same i-node number	find . –inum {inode #}
Rename all K & S Scripts to disable them:	mv /etc/rc2.d/S88sendmail /etc/rc2.d/xS88sendmail

- Remove any software that is not being used
  Even better than disabling is removing any software that is not required all together. Any software that is installed, but not running, can pose problems from a security point of view. While an intruder cannot use idle software to gain access to the server, it can be exploited in a number of ways. For example, they could use a compiler (resident on the system from compiling current service binaries) to build additional hacking tools on the system. These could then be used to gain further access to additional systems. Or, the intruder could plant a backdoors in an unsuspected location, like an idle service. Administrators aren't likely to examine services that aren't currently in use for trojans, making them ideal targets – just in case the system ever starts using them again. By removing software that is not in use, you make it more difficult for the intruder.

  How difficult it is to remove software is directly dependant on how it was installed. If the software was compiled and installed via a make script, or some similar install script, you are usually left to do the uninstall by hand – manually deleting the file. In this case, it can be quite difficult to track down all the individual pieces to the software, though usually simply deleting the application binary – while not clean – will suffice from a security point of view. The other way that Solaris allows for software to be installed is through *packages*. To identify which packages are installed on a system, you use the *pkginfo* command. This provides you with a list and a description of the software. Keep in mind that the majority of the operating system is installed as packages, so if a full install of Solaris has been completed, this list can get quite long. To remove a package, you use the *pkgrm* command. Launched without any command line options, this command will allow you to interactively select the patches you want removed. However, if you happen to know the name of the package you want to remove, you can specify it on the command line for a much quicker remove.
- Consider running a Port Scanner on a Regular Basis
  Many scanners (such as nmap – discussed later), are easily downloaded and compiled. Run these programs, on a regular basis, against the servers in question to determine if they are responding to queries that you may not be expecting them to be. By performing regular scans, you can catch services that may have been restarted to fix a one-time-problem (and were not shut off again) or were restarted by a patch that was applied. I would recommend this be done anytime you apply a patch (or set of patches) to the server.
- Audit Servers on a Pre-Determined Schedule
  By performing regular audits on the servers, you gain 2 benefits. First, you insure that the server is not running any of the services that it should not be. Secondly, the administrator becomes more acutely aware of what is supposed to be running and will notice quicker if something is out of place.

Permission & Ownership Problems

Fix-Modes Script
http://www.sun.com/blueprints/tools/FixModes_license.html

Sun is now providing access to a tool written by Casper Dik (Casper.Dik@Holland.Sun.COM) that examines your current operating system and "locks down" the default file permissions of a base Solaris install in an effort to improve the general security of the system. Since the tool uses the contents of */var/sadm/install/contents* to determine which packages have been installed on the system, any application that is not installed with the *pkgadd* command won't be repaired. (This would include applications, like fix-modes itself, that are compiled on the system and installed using make or similar install scripts. It is highly recommend that this script be run on any newly installed machines, however care should be exhibited on servers that are currently in service.

SUID/SGID Problems

The *Set User ID* (SUID) and *Set Group ID* (SGID) bits behave in almost the same manner. The majority of the time, these permission bits are set to allow users to run programs/scripts with another users' ID. In the case of the SUID bit, the script runs as the user who owns the file – not the one who is executing it. Similarly, a SGID script runs as the group who owns the file. This is particularly important when considering what access those scripts could be granting.

A SUID/SGID script is identified by an "s" in the execute column of a file's permissions. Take /usr/bin/passwd – you can see the "s" in both the user and group execute columns. This signifies that this script has been set both SUID and SGID.

```
-r-s--s--x   3 root     sys        85652 Nov 21 15:03 /usr/bin/passwd
```

Now why are SUID and SGID necessary? Continuing with our example of /usr/bin/passwd, we are all aware, any user who wishes to change their password uses this program. In order to modify /etc/passwd and /etc/shadow, the program requires root level privileges. To get around all users logging in as root to change their passwords, the passwd program is set as SUID. This tells the program to run as whomever owns the, and in this case, that is root.

Another use for SUID and SGID bits is to set them on a directory. This forces a file to be created with a particular user or group regardless of the user/group creating the file. (As long as the directory permissions themselves allow for the file creation.)

Let's say we have a directory that we use to share files with colleagues. Let's also assume that the majority of the users have this "group" as their secondary group. Take the (fictional) user *pooh* – who has the following identification (obtained by doing an *id –a pooh*) on the server:

```
uid=1001(pooh) gid=10(staff) groups=100(bear)
```

Let's assume we set the directory /data/woods/100acre to have the following permissions:

```
-rwxrws---   3  chris     bear        512   Nov 21 15:03 /data/woods/100acre
```

Now, none of the users would have to force the group of their files deposited in the 100acre woods as the SGID bit automatically sets this ownership. It's also worth mentioning that while the SGID bit is inherited by subdirectories created under a SGID directory, the SUID bit is not. Thus, if a parent directory has both SUID and SGID bits set, a subdirectory created would only have it's SGID bit set.

SUID and SGID scripts only become a serious security concern when a program is SUID root, like /usr/bin/passwd. The reason that this comes as a concern is that these files are now running as the root user. Should a hacker determine a way to interrupt the execution of this program, it is quite possible that they might be able to obtain a root prompt.

To find all of the SUID root programs/directories on the system, we ran the following find command:

find / -user root –perm –4000

The list produced was all the SUID files that are distributed with the Solaris operating system. Due to the limited access of 'real users' to these systems, you may be able to get away with removing some (but not all) of the SUID access to these files and granting individual users access to certain programs through *sudo*, however I would be quite wary with making changes to the default SUID programs distributed with the operating system.

System & Application Logging

An analysis of the configuration file for *syslog* from *panther* shows us that information is being saved locally. When administrators are queried as to why they have things configured this way, the only justification they could give was that this is the way it has always been. Since it's not broken, they see no reason to fix it. While this configuration is not bad, since it insures no data loss due to network failure, it does pose a problem with respect to the security of the logs. Should an attacker compromise the server, it is quite easy for these logs to be modified to erase the hackers' tracks. Thus, we suggest that in addition to the resident logfiles, that logging also be sent to a dedicated "log server". This server should reside in a secure portion of the GIAC Enterprises network – a recommended location would be in a similar location to the database server, *lynx,* which is protected by a Checkpoint Firewall.

Before looking at sending the messages to the protected log server, insure that the local files are as protected as they could be. Examine the ownership and permissions of the /var/log directory, and all of the logfiles within, to insure that only root has read/write access to these files. There is no point in allowing "prying eyes" to be able to easily see what is being logged locally.

To instruct the syslog daemon to begin sending logs to the central log server, you will need to modify your syslog's configuration file (/etc/syslog.conf).

Existing /etc/syslog.conf on *panther*:

```
*.err;kern.notice;auth.notice                  /dev/console
*.crit;kern.err;daemon.err                     root
*.emerg                                        *
kern.debug                                     /var/log/kernlog
```

```
user.info                                       /var/log/userlog
mail.info                                       /var/log/maillog
daemon.info                                     /var/log/daemonlog
auth.info                                       /var/log/authlog
news.info                                       /var/log/newslog
uucp.info                                       /var/log/uucplog
syslog.info                                     /var/log/syslog
```

The modification is quite simple – for each line in the existing syslog.conf, add another that directs the same content to the log server for archiving. For example, the first line would be replaced with the following 2 lines:

```
*.err;kern.notice;auth.notice                   /dev/console
*.err;kern.notice;auth.notice                   @loghost
```

Notice that we used the @ notation in our new syslog.conf to define a remote destination for the log information. Since we use @loghost, this instructs the syslog daemon to look for a server called 'loghost' and sends the log information to syslog running there. Usually, the name of the log server is only defined in the system's host table (/etc/hosts) but it can also be defined in DNS. By default, the system has defined *loghost* in /etc/hosts to be the server on which it reside. Our new configuration will require that entry be modified to reflect our new server. We do not recommend using DNS to define your loghost, since it makes the log server much easier to identify.

After any change to the syslog.conf, you must send a hang-up (HUP) to the syslog daemon for the changes to take effect.

Currently, there is no process in place for the rotation of system logs and they are manually purged when the /var file system is nearly filled. We recommend that a process be put in place to rotate the system logs on a regular basis (daily or monthly, depending on the space requirements and availability) and have them moved to a remote server for archiving. These archived logs should then be periodically stored on CD-ROM (monthly or yearly, depending on rotation frequency and space requirements). A similar process should be put in place on the log server so that the logs from the central log server can be compared against the server's local logs to detect tampering. By keeping these logs archived on CD-ROM, it provides easy access to older logs in the event of a system compromise where archived logs may need to be examined.

It's worth mentioning that should the server be compromised, the attacker could simply modify the syslog.conf to disable the remote logging – or logging all together for that matter. However, with the logs on the central log server, there will hopefully be a trace as to how the hacker got in to the server and while he can erase that evidence locally, he will have a more difficult time getting at the archived logs on the log server.

The web server's access and error logs already go through a rotation/archive process similar to that which we propose for the system logs. The reason that this process is already in place is due to analytical requirements of this data for marketing trends and capacity management.

As defined in Apache's configuration file (httpd.conf), the web server logs to the directory /data/logs/www/, creating an access (giac-access) and an error (giac-error) log – the directives shown here:

```
TransferLog /data/logs/www/access-giac
ErrorLog /data/logs/www/error-giac
```

Both access and error logs are rotated nightly – as compressed, dated files - and stored in /data/logs/www/archive for pickup by the log archiving process. For example, the access and error logs rotated on November 29th, 2001 would be renamed to access-giac.20011129.Z and error-giac.20011129.Z and stored in /data/logs/www/archive. An automated process running on a dedicated log-processing server (possibly a candidate for the central syslog server required above?), collects the rotated logs from the remote web servers. On this server, the logs are archived and analyzed for capacity management and market/customer trends. Only the current year's logs are available on the log processor – once the year end statistics are collected, logs are copied out to removable media (currently CD-ROM) and then stored offsite.

The last set of logs that are crucial for tracking access to the server is that of the transaction logs from the ftp server. The wu-ftpd currently installed is configured to log to /data/logs/ftp/xferlog and this is not something that is easily changed as it is a compile-time option. Currently, this log file is treated much like the syslog files in /var/log and are simply purged whenever the /data filesystem nears capacity. These logs should be rotated/archived in a similar manner to that of the web logs described above – though potentially on a different schedule. It is important to keep records of all transactions to the server in case there is ever a dispute as to how a particular file got onto the server.

Major Application Configuration

Sendmail

At a glance, you would not think that a web server would require a mail server. This is only partially true – a web server does not need to have a mail server continuously running. Whenever a program sends mail – be it a CGI script sending feedback from a web page or an administrative script delivering daily server statistics to an administrator – if sendmail is not running, it will start up and attempt to deliver the mail.

Since there is no need for us to receive e-mail on this server – this would be the only reason you would configure sendmail to run continuously – it is much easier for us to simply shut the program off following the steps covered in Running Unnecessary Services rather than attempt to securely configure the application.

However, now we have to take into consideration the odd time that mail will not automatically get sent out due to a problem with the destination mail server or a network problem. In this case, the web server will queue the message for later delivery. Unfortunately, with sendmail disabled

the queue will never get reprocessed. To get around this problem, simply schedule a cron job to process the queue every hour. The entry in root's cron should be similar to the following:

```
0 * * * *  /usr/lib/sendmail -q
```

WU-FTP

FTP services are provided on *panther* to allow system administrators and web content developers to move files to and from the system. In both cases, the processes using ftp are a mixture of automated processes (like the web log rotation process discussed above) and manual file transfers (administrators moving new application binaries to the server). Due to the limited user base, there are some areas where access to this service can be restricted.

Overall, we were quite pleased with the configuration of the ftp service on the web servers. Currently, only real and guest users have both been defined in the *ftpaccess* file, which only permits "real" users to log into the server. No anonymous logins are permitted and have been disabled using the configuration *defaultserver private* in the ftpaccess file. Web developers, which belong in the "webdev" group are defined as "guests" using:

```
guestgroup webdev
```

where *webdev* is a valid group in /etc/groups. System administrators are considered to be "real" users, and are of defined using:

```
realgroup admin
```

Likewise, *admin* must be a valid group defined in /etc/groups. In the case of the web developers, they can only belong to the webdev group – a limitation of "guest" ftp accounts, but the system administrators can be members of multiple groups, as long as their primary group is "admin". For the access required by the web developers, this suits their needs.

In addition, some precautions have been taken to protect some of the system files using the *noretrieve* option. Some of the files protected by this rule are *passwd, shadow,* and *.rhosts*. It is no surprise that these files should be protected from being downloaded from the ftp server.

with respect to the files being uploaded to the server. In the ftpaccess file, users who upload content to the web site, the file ownership and permissions are pre-determined by the ftp server using the following configuration line:

```
upload /data/www /html yes www webdev 0660 dirs
```

This instructs the ftp server to create all files uploaded into the web tree (since this directive will apply to /data/www/html/*) to an ownership of user *www* and group *webdev* with permissions of 660 (rw-rw----). Default umasks have also been set:

```
defumask 0027 real
defumask 0027 guest
```

This creates files uploaded outside of the www directory to be created as 750 (drwxr-x---) dirs and 640 (-rw-r-----) files.

Of course, if you examine one of the web developers accounts, you'll see that it has been chroot'ed (indicated by a "." in their home directory path) into the web content directory anyway:

bbailey:x:1001:10:Beetle Bailey (Web Developer):/data/www/./html:/usr/bin/passwd
Also note that the user's shell has been set to /usr/bin/passwd. This insures that this particular user cannot actually log in to the server, but is able to change the password when the system expires it.

The last configuration option worth mentioning is that, the ftp server has been configured to disallow the "guest" user to change their umask and the permissions of files through their ftp client via the following parameters:

umask	no	guest
chmod	no	guest

There is always some concern when you allow an ftp service access to "live" web content directories, as they currently have configured. However, we believe that the risk presented to GIAC Enterprises is minimal given the limited use of the product. Once the recommendations in this section have been implemented, we feel that this service will be adequately secure.

We were surprised to find wu-ftp unrestricted by TCP Wrappers, given the fact that Wrappers is already restricting access to the telnet daemon. In conversations with the administrators, this is a legacy configuration that was required when the company used to contract out the creation of their fortune cookie sayings. Their consultants would use the ftp service on *panther/leopard* to upload new content. Under current operations, this content is now developed in-house and thus this configuration is no longer required. The first thing that we recommend is to put wu-ftpd under control of the TCP Wrapper daemon. The section on the configuration of TCP Wrappers will provide more information and examples of the inetd.conf entries.

The other thing that will enhance the security of the server, not just with ftp but with most networked services, is to insure that connections originating from within the GIAC Enterprises LAN be forced to communicate along the management plane for these servers. This will protect both the ftp and telnet/ssh services which should only be answering on the management side of the service. If a potential hacker somehow discovers the internal network IP's and then spoofs a connection, static routes will force the return communication out the management plane rather than the service plane (where we assume the connection attempt came from) and the session will be killed by anti-spoofing rules on the Checkpoint firewall.

During the recommended rebuild of the web servers, administrators should insure that they install the current stable release (2.6.2) of the ftp server software. As discussed earlier, the version currently in production (2.6.0 (Beta 1)) is vulnerable to exploit. See www.cert.org/advisories/CA-2000-13.html and www.cert.org/advisories/CA-2001-33.html for more information.

Apache

Since we are auditing web servers, naturally these servers must have some sort of web server software installed. GIAC Enterprises is currently running the Apache Web Server (www.apache.org/httpd), which is the #1 web server on the Internet, according to the Netcraft Web Site Survey. (www.netcraft.com/survey/)

We will summarize some of the configuration options below, most of which have already been set on *panther*, however our biggest concern with respect to the web software resides in the fact that the particular version (1.3.4) is currently several revisions behind. We suggest that, during the recommended rebuild of the server, that the most current stable version (1.3.22) be installed.

The first set of directives, set in the Apache configuration file (httpd.conf), we are going to look at will determine which portions of the UNIX filesystem is available through the web server. Since this is a public website, we will want to grant unconditional access to the HTML content, while restricting access to the rest of the UNIX system. This is accomplished with 2 *Directory* configurations:

Deny access to the entire system

```
<Directory />
        AllowOverrides None
        Order Deny, Allow
        Deny from all
</Directory>
```

Grant access to the HTML document root

```
<Directory /data/www/html >
        AllowOverrides None
        Order Deny, Allow
        Allow from all
</Directory>
```

Obviously, the important lines here are the first *Directory* lines, which define the scope of the following configuration options and then the *Allow|Deny from all* lines. These directives can be repeated for any directory that visitors require access. By denying access to the entire system first, and then only allowing certain portions access, you are able to maintain tighter control of who goes where. If someone tries to trick the web server into displaying content outside the allowed directories – in our case, outside of the HTML document tree - they will be stopped.

The next parameter we noted was the AllowOverride, which is currently set to "None" for all directories within the web tree. This disables the ability to override the default configuration of the web server in a special file called .htaccess. Disabling this parameter is a very good configuration, for 2 reasons.

First is that it will boost the performance of the web server. Every time a item is requested from the web server – html document, image file, movie file, etc - it will not be presented to the client until the system insures that there is no .htaccess file present that might modify how it handles that particular request. The problem is that the system does not only search the directory housing the content, but searches the entire tree – from the document root down to the directory housing the content. The reason for this is that .htaccess files override the options recursively – thus an .htaccess file in the document root applies to the entire website.

The second reason that this is a good configuration is because the web developer can no longer override the configuration set out in httpd.conf. Let's say, for example, that you have disabled CGI script execution in a particular directory (discussed later) – the user would be able to override that decision and re-enable the execution of code through a .htaccess file. Obviously, if you set out strict security rules in your configuration files, you do not want the user to be able to bypass them at their leisure.

The next configuration option we were particularly pleased with was the configuration of a restricted access "CGI" directory.

```
Define where your CGI directory is      ScriptAlias      /cgi-bin/  "/data/www/cgi-bin/"
Grant access to the CGI dir, but        <Directory  "/data/www/cgi-bin/">
 remove all options                             AllowOverride None
                                                Options None
                                                Order allow,deny
                                                Allow from all
                                        </Directory>
```

This allows the web server to access to a cgi script by using the web path of /cgi-bin/*scriptname* where *scriptname* is the program being executed. When the server is instructed to retrieve a file from /cgi-bin, the *ScriptAlias* directive tells it to look for this file in /data/www/cgi-bin/ rather than a directory within the document tree. Access to this directory should be controlled tightly, keeping in mind that the web server user has to have the ability to run the scripts.

We some additional that are set also enhance the security of the web server - such as having directory indexing turned off and the use of IncludesNoExec to stop scripts from being included in web pages. With indexing turned off, it prohibits visitors from viewing the contents of a directory which does not have an index file (usually index.html). The "IncludesNoExec" parameter allows the web developers to use all Server Side Includes (SSI) except the execute (exec) directive which allows for the execution of arbitrary code as the web server.

The biggest surprise that we had was the discovery that GIAC Enterprises is conducting financial transactions without the use of Secure Socket Layer (SSL) encryption. Since clients are entering credit card information into various forms on the website, those portions should be secured with an encrypted tunnel. In order to enable SSL encryption on the web server, the Apache server has to be compiled with "mod_ssl" (www.modssl.org), an option that was not included with the current version of Apache. When building the new version of Apache, as suggested above, we highly recommend that mod_ssl be included at that time as well. In order to build mod_ssl, you will also need OpenSSL, which is also a requirement for OpenSSH (discussed later).

When deploying the SSL-enabled website, you may want to consider running it as a separate web instance, since SSL is enabled at the site level, not on a per-directory level. (In other words, either a site is conducting all transactions encrypted or unencrypted.) Websites will use 2 identical web instances, pointing at the same content, to give the impression of being able to enable encryption on a directory-by-directory basis. They simply switch back and forth between the http and https websites as necessary.

Once the new SSL-enabled Apache is built, you will need to configure the web server to conduct transactions using SSL. An example of how to enable the security on a single directory is shown here:

```
<Directory /data/www/shtml/>
        SSLRequireSSl
</Directory>
```

If a non-SSL connection is attempted to a directory protected by this directive access will be denied. Additional examples of how SSL can be configured on the server can be found in the mod_ssl documentation.

Please keep in mind that we are only encrypting the transaction between the Client PC and the Web server. When the CGI Script on *panther* established the connection to the database server inside the GIAC Enterprises' network, the content is sent in plain text. In discussions with server administrators, they tell me that the script storing the information into the database server is storing the information encrypted. We were never able to independently verify this claim, but all supporting documentation appeared to support this claim. Since this transaction is completed through a secure network connection, it is of little concern.

Known Vulnerabilities

As mentioned in the section Administrative Practices, by keeping your server patched and subscribing to vulnerability mailing lists you can keep on top of things with new exploits. However, what about the existing problems in the software you run? By searching through the archives for the vulnerability mailing lists, you can look for any that pertain to the software you are running. Also, pay close attention to websites like the SANS Top Vulnerabilities which will point out key problems with common applications.

To test the current configuration of *panther*, we ran 2 tools from outside of GIAC Enterprises' network. The first test we ran was to do a port scan of the server using *nmap* (www.insecure.org/nmap) and the second was to do a vulnerability scan using *nessus* (www.nessus.org). It was important to launch these scans from outside of the corporate network so that our experience would be closer to that of an attacker's. The output from these commands can be found in the Appendix.

Overall, the servers faired quite well against these scans given the fact that we know that they are indeed running extra services. Nessus did detect some items that it labelled as security warnings or security notes. Each entry in the list has been prioritized as to it's severity and nessus even provides a link to solutions, should they exist. There are two services that we were surprised were allowed through the PIX Firewall – those being telnet (port 23) and ftp (port 21). Given the statement about ftp being required from the internet in the past, the fact that ftp was still being allowed through did not surprise us as much as telnet traffic being passed. There is no immediate threat to telnet as both TCP Wrappers and static routing on the server protect this service, however the wu-ftp service is completely unprotected and running with documented exploits! Again, we reiterate the necessity to reconfigure TCP Wrappers and we strongly

suggest that port 21 be closed in the PIX Firewall to protect this service – at very least until the ftp server binary is updated.

Comments:

- Keep abreast of Common Internet Security Vulnerabilities
  Monitor some of the more popular websites that discuss which security vulnerabilities are currently 'en vogue'. For example, SANS maintains a list of the 20 critical Internet security vulnerabilities – a list that is determined by a consensus of industry experts. It is a good idea to watch this list because it will give administrators an idea what the most commonly exploited holes are. Do not limit yourself to "White Hat" websites as well – encourage your administrators to read some of the common hacking websites. (i.e. www.rootshell.org) Provide your administrators a way to easily share useful websites that are discovered – perhaps a web page on an internal "support" web server.
- Consider running a Vulnerability Scanner on a regular basis
  Testing the integrity of a server is a good way to know if it is as secure as you think it is and the best way to do that is by using a vulnerability scanner. One of the more popular scanners – used by hackers and security administrators alike, is a program called Nessus. It can determine how well your existing security policy can deal with known attacks. Most importantly, the scanner will point out the weak points, which you can then strengthen. These scanners are also good ways to test your firewalls and Intrusion Detection Systems (IDS) as the scanner will, in most cases, actually employ the vulnerability. Running a vulnerability scanner, which is kept up to date, on a regular basis, will periodically verify that your server is still as secure as you had thought is was, as well as probe for new exploits discovered since the last run.

  As part of ongoing policy, we suggest that any new machines being deployed into the network be subjected to a security "stress test" by scanning them with these tools set to their most intrusive levels. This would expose any weaknesses prior to them being deployed to production service and allows a consistent testing of current security standards.

  Care should be taken whenever using nessus against a production server as a number of the scans included with nessus are considered "dangerous" and could crash the server being scanned. Appropriate measures should be taken to insure minimal impact to the customer.

System Monitoring & Response

System Monitoring takes on 2 purposes – first to alarm on abnormal behaviour/hacking attempts and the second to insure that the service you are providing is functioning the way you expect it to be. To this extent, GIAC Enterprises only partially meets these requirements. Using HP's IT Operations, the company monitors for functionality and limited abnormal behaviour. (For example, they monitor for incorrect login attempts; make notes of who switched to the super user) There is plenty of abnormal behaviour that could slip through the limited scanning of system logs (if they are even logged).

GIAC Enterprises policy for dealing with system alarms is well documented and the administrators interviewed were well versed in their expectations. The policy document is reviewed by key stakeholders on a regular basis and is kept current such that most scenarios are covered.

Comments:

- Consider configuring ITO to page on all "unrecognised" behaviour
  By configuring the network monitor to page on unrecognised patterns – not just things that it knows about – you can greatly cover what information you are capturing. Keep in mind that configuring things in this methodology can generate a significantly large number of "false positives"; at least during the initial deployment until the majority of the alarms have been identified.
- Consider deploying a Network Intrusion Detection System (IDS)
  Traditional IDS systems monitor network connections for recognised abnormal behaviour and can act on it. How the IDS responds depends on which software you are using, but can range from dynamically reconfiguring a firewall to block access from that port or IP, to sending off a page to notify an administrator. Engage the appropriate people to evaluate the software available and deploy.

Additional Software Recommendations

Host Filtering Software (TCP Wrappers)

If the question was put to us, if we could choose only one recommendation from our report to implement, we almost always will respond with the installation and configuration of TCP Wrappers. Fortunately, GIAC Enterprises already has partially deployed an older version of this software on the web servers *panther* and *leopard*. We'll take this opportunity again to strongly suggest that the wu-ftp service be included in the services protected by TCP Wrappers.

The software is free and can be downloaded from ftp://ftp.porcupine.org/pub/security/ This site should be monitored for new revisions and bug reports. Since this product is one of your frontline defences, it is imperative that the software is kept up to date and bug/exploit free. (As much as possible.)

As mentioned earlier, we are dealing with quite a small user base with the GIAC web servers. System administrators use telnet and ftp for remote management and maintenance of the servers and web developers upload content to the web server using ftp. Since there is only a need for telnet and ftp services, these should be the only two lines left uncommented within /etc/inetd.conf (see Running Unnecessary Services). Since both of these groups will only be accessing the system from within the GIAC Enterprises' corporate network, TCP Wrappers should be configured so that only certain hosts or network segments can connect to the server. In examining the configuration files on *panther,* this is exactly what is has been done.

Telnet & ftp entries from *panther's* /etc/inetd.conf:

```
telnet  stream  tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd
ftp     stream  tcp nowait root /local/sbin/in.wuftpd in.wuftpd -a -u022
```

As you can see, the telnet entry is calling *tcpd* (the TCP Wrappers daemon) before it executes the telnet daemon. The ftp entry, calling wu-ftpd, is not. To secure wu-ftp, the ftp entry in /etc/inetd.conf needs to be changed to:

```
ftp   stream  tcp nowait root /usr/sbin/tcpd /local/sbin/in.wuftpd -a -u022
```

Crucial to the successful configuration of TCP Wrappers are the hosts.allow and hosts.deny files. These are the files that allow tcpd to determine if a particular connection is permitted or not. The theory used in most implementations of TCP Wrappers is similar to that of the configuration of the web server above – deny access to everything and then allow access as required.

By examining the hosts.allow and hosts.deny files from *panther*, we quickly see that this is the way that GIAC Enterprises has configured TCP Wrappers.

hosts.deny

```
ALL: ALL: banners /local/etc/TCP_Wrappers-Msgs/deny
```

hosts.allow

```
# System Administrator Telnet Access
in.telnetd: IP Addresses Censored: banners /local/etc/TCP_Wrappers-Msgs/allow
```

In order to complete the protection of wu-ftpd, we will add 2 additional lines to hosts.allow:

```
# System Administrator FTP Access
in.wuftpd: IP Addresses Censored: banners /local/etc/TCP_Wrappers-Msgs/allow
# Web Developer FTP Access
in.wuftpd: IP Addresses Censored: banners /local/etc/TCP_Wrappers-Msgs/allow
```

Notice the *banners* directive in these files. It instructs *tcpd* to look in that directory for a file named after the service (i.e. in.telnetd) to display when it matches that particular rule. For example, upon making a successful telnet connection, TCP Wrappers will display a file called /local/etc/TCP_Wrappers-Msgs/allow/in.telnetd. Likewise, upon denying a FTP connection, it will display a file called /loca/etc/TCP_Wrappers-Msgs/deny/in.wuftpd.

Secure Shell (OpenSSH)

On the surface, an interactive session – usually a telnet session or remote shell (rsh) - between your desktop computer and a server seems harmless. Unfortunately, a much closer examination of the communications between these two machines (using, for example, a snoop command or some other packet sniffer) can produce some surprising results. All information being sent is done with no encryption what so ever – so things like usernames and passwords are easily intercepted and stored by "curious" users.

Unfortunately, this is the way that system administrators at GIAC Enterprises access their systems. Deployment of Secure Shell (SSH) will easily resolve this. Even in it's most insecure

mode, SSH has established an encrypted tunnel (much like the web server's secure socket layer) between your desktop and the remote server prior to you typing in your username and password (or the root password, for that matter.) The encrypted tunnel would make capturing this information virtually impossible, as the hacker would no longer be able to easily pick out the username/password data from the data stream.

OpenSSH for Solaris is available from http://www.openssh.com/portable.html. Since OpenSSH is developed primarily for use in the OpenBSD Operating System, we must use code that has been "cleaned up" by the OpenSSH portability team. The newest version, as of December 20th, 2001 is 3.0.2 that offers support for both the SSH 1 and SSH 2 protocols.

As we mentioned above, SSH in it's most insecure setting simply establishes an encrypted tunnel, using host (rather than user) based public key authentication. In this situation, the client and server exchange keys to establish the encrypted tunnel, and then the .rhosts and/or hosts.equiv files are examined to determine a user's ability to gain access to the server. Since we want to discourage users from using these files (see Operating System Configuration), we really do not want to do this sort of authentication. SSH can also be configured to implement user (rather than host) based public key authentication. There is more administration involved with this sort of authentication, as each user's public key must be generated and then copied over to the server they require access to. In addition, any time that a user modifies their key; the copy on the server needs to be updated as well. Even with all this extra work, the user-based public key authentication is the superior method as each individual user's access is identified by the presence of their public key on the server.

Before installing OpenSSH, you will need to have installed OpenSSL (already required for building mod_ssl into Apache – see Major Application Configuration) and TCP Wrappers. The reason for this is due to the options for building the OpenSSH software:

```
--with-ssl-dir=/path/to/the/OpenSSL/library
--with-tcp_wrappers
```

The *with-ssl-dir* option instructs the OpenSSH build process where to find the OpenSSL library files. The *with-tcp_wrappers* enables OpenSSH to use the host.allow and host.deny files, in the same manner as TCP Wrappers does. To insure an easy build, you should create a link in the OpenSSH build directory to wherever tcpd.h is located on your server. While *with-tcp_wrappers* is not required for the OpenSSH build, it is highly recommended.

Once the binary is built, what type of authentication OpenSSH uses is configured in /etc/openssh/sshd_config. As mentioned above, the software does support using .rhosts / host.equiv files for authentication, however we do not want to use this method. In the software's default configuration, this behaviour is disabled, but you should insure that your *ssd_config* file contains the following lines:

```
IgnoreRhosts    yes
RhostsAuthentication  no
RhostsRSAAuthentication     no
```

It is likely that, should the decision be made to switch to user-based public keys, the implementation will take time. To continue allowing people to authenticate with

username/password pairs, you need to insure that the following 2 lines are incorporated you're your *ssd_config* file:

```
PasswordAuthentication      yes
PermitEmptyPasswords        no
```

After the software has been installed and configured, administrators will need a copy of the ssh client (distributed as part of the OpenSSH package) installed on their workstation. To establish a connection to a remote server, they call the command as follows:

```
ssh –l lavander panther.giac-enterprises.com
```

If the administrator is unlucky enough to have a workstation not running some flavour of UNIX, there are SSH version1 and SSH version 2 clients available for a variety of other operating systems.

Sun has developed a document to assist in the installation of OpenSSH on the Solaris platform. It can be found here: http://www.sun.com/blueprints/0701/openSSH.pdf

Host Intrusion Detection Software (Tripwire™)

Tripwire is software that allows you to detect changes, no matter how small, with selected files on your system. A code (called a checksum), is calculated based on the contents of the file – thus the checksum will change should any of the file's contents change. Tripwire operates in two modes – database creation and system scan. If you run the database creation function of the software immediately after the system has been build (clean operating system install and all required applications), your database will now contain a baseline to which the current checksums can be compared. In the system scan mode, it does exactly that and produces a report of all files that have different checksums than the baseline database.

Tripwire is good for keeping track of static operating system files, but poor at monitoring files that are frequently modified. Commonly accessed commands, like passwd and ls, are unlikely to change but are prime targets for hackers to replace with trojan versions. By having Tripwire monitoring your files, it will allows you to detect the trojaned versions quickly, hopefully before they have time to do any significant damage to the company.

Tripwire (www.tripwire.com) is available in several different forms and different platforms. In addition to a variety of operating systems (including Linux, several UNIX variants, Windows NT/2000), specialized versions of Tripwire are also available for monitoring Router/Switch files as well as Web Content. While we encourage GIAC Enterprises' to explore the possibility of deploying the commercial versions of Tripwire across the organization, we also recognize the cost involved with such a project. With that in mind, we recommend (at very least) the deployment of Tripwire AR, which is available free of charge. While it is not as full featured as the commercial versions (and can only be used on servers), it will establish a valid change monitor for the files and directories on the GIAC web servers.

The first thing that needs to be done after the installation of the software is to decide which files and directories to monitor. As mentioned above, files that do not to change on a regular basis are the best candidates. Exclude entire directories (like /var/log) when you can get away with it, but

be careful what files you are accidentally letting through the cracks. The /bin and /lib directories are obvious targets for hackers, since the operating systems' binaries and library files can easily be replaced with trojaned versions. The /dev directory is a favourite for hackers to 'hide' their tools, and should not contain files that change frequently. This directory is not frequently visited during daily administrative tasks and tends to contain a lot of 'odd' named files.

Since we are concerned with GIAC Enterprises' website being modified by a hacker, we recommend that Tripwire be allowed to monitor the web directories for changes. Since the content of the website does not change frequently and when changes are made, they are coordinated with the system administrators. This would allow the Tripwire database to be updated after the web content had been updated. Another time you will have to rebuild the Tripwire database is after you have applied operating system patches. There is a good chance that one of the files you are monitoring will have been modified by one of the patches that are applied and thus, the new file will have a checksum that is different than the one that is on file.

After the administrator chooses which files to monitor they must be entered into the tw.config file. From the manpage for tw.config, here is an example:

```
/etc            R       # all system files
!/etc/lp        R       # ...but not those logs
=/tmp           N       # just the directory, not its files
```

Each line in the tw.config file determines exactly what is going to be monitored. The parameters allow the administrator to add or remove detection of changes quite easily. It also allows the admin to choose the type of checksums to run against the listed files. Again, from the tw.config manpage, the full lists of options are as follows:

p	permission and file mode bits
i	inode number
n	number of links (i.e., inode reference count)
u	user id of owner
g	group id of owner
s	size of file
a	access timestamp
m	modification timestamp
c	inode creation/modification timestamp
0	signature 0 - null signature
1	signature 1 - MD5, the RSA Data Security, Inc. Message Digesting Algorithm.
2	signature 2 - Snefru, the Xerox Secure Hash Function.
3	signature 3 - CRC-32, POSIX 1003.2 compliant 32-bit Cyclic Redundancy Check.
4	signature 4 - CRC-16, the standard (non-CCITT) 16-bit Cyclic Redundancy Check.
5	signature 5 - MD4, the RSA Data Security, Inc. Message Digesting Algorithm.
6	signature 6 - MD2, the RSA Data Security, Inc. Message Digesting Algorithm.
7	signature 7 - SHA, the NIST Secure Hash Algorithm (NIST FIPS 180)
8	signature 8 - Haval, a strong 128-bit signature algorithm
9	signature 9 - null signature (reserved for future expansion)

R	[R]ead-only (+pinugsm12-ac3456789) *(default)*
L	[L]og file (+pinug-sacm123456789)
N	ignore [N]othing (+pinusgsamc123456789)
E	ignore [E]verything (-pinusgsamc123456789)
>	monotonically growing file (+pinug>-samc1233456789) - the ``>'' indicates that file changes are ignored only when the file is

```
smaller than the last recorded size.  This is useful for log files
that are expected to grow.
```

The default mode (Read-only) should suffice for most files being monitored. However, if Tripwire appears to be running slow, you should examine your tw.config file to see if you can trim back the number of checksums you are calculating on a particular file (by default, you calculate both the MD5 and Snefru checksums). If you are having performance problems, we suggest that calculating the MD5 checksum alone would also be acceptable.

As mentioned above, after the initial "baseline" database is created, Tripwire needs to be run on a regular basis. We recommend scheduling a daily run of *tripwire –q* in root's crontab. When Tripwire is set to scan, the software examines the tw.config file to determine what data needs to be compiled for comparison against the database. If Tripwire detects any changes to a monitored file, it will result in the application sending an e-mail to the administrators. We also recommend that, from time to time, Tripwire be run manually outside of the regularly scheduled scan. This sort of scan enables you to catch attackers "off guard" since the scan isn't supposed to run at that time. As well, since the administrators are running the application manually, they will be able to insure that the application is functioning and reporting properly.

We believe that this software is key to the detection of a compromise. Unfortunately, to properly implement this service with any level of confidence, it cannot be deployed on an existing server. As a result, this software should be deployed when the servers have been rebuilt as suggested in **Added Security through Better Initial Configuration.**

Superuser DO (SUDO)

At the moment, GIAC System Administrators all have access to the root password through a centralized password database. Passwords are stored within this database to allow any system administrator to log on to any servers as root. While we understand why the database exists, it does present a security risk. To overcome this, we suggest 2 steps be taken.

First, develop a hard copy of the root passwords that is stored in a secure location – like the vault where tape backups are currently being stored. Insure that this hard copy is kept up to date as it will most likely only be used in a time of emergency.

Second, remove the requirement for system administrators to be root to accomplish their daily tasks. The easiest way to implement this suggestion is through the deployment of a program called *sudo*, short for **su**peruser **do**. This command allows you to run configured commands as another user, usually root, after supplying your login password. The current (stable) version of *sudo* is 1.6.3p7 is available from www.courtesean.com/sudo/ftp.html, however the *sudo* website reports that version 1.6.4 will be available January 2002.

Some of the features provided through the deployment of *sudo* are:

- Increased accountability for individual administrators as all commands executed through *sudo* are logged.

- Removed dependency on the root password (for specific individuals) to perform routine administration tasks.
- Access privileges can be changed , or even revoked entirely, without changing the root password. This removes the necessity to change the root password on all systems when an employee leaves the group, though this practice would still be strongly encouraged.
- A single configuration file can be created for use on all servers. In addition to making it a quick configuration on subsequent installs of *sudo*, the configuration file will also provide administrators with a list of which user has what access.

A word of caution – should a *sudo*-enabled user account be compromised, the intruder would be granted the same permissions as the user. However through significant network security, simple social engineering, and good (and frequently changed) user passwords, this risk is considered to be minimal.

Configuration of the application is done through */etc/sudoers*, which is edited using *visudo*. This is a special version of the vi editor that does some syntax checks before allowing you to exit. A basic *sudoers* file consists of 4 major sections:

- Host Aliases
  This section is used to give a "simpler" name to either a single or multiple servers.
  For example:

```
Host_Alias      HOME=myserverathome.mynetwork.com
Host_Alias      KITCHEN= toasteroven, waffleiron, microwave, wok
```

- User Aliases
  This section is used to give a "simpler" name to either a single or group of users.
  For example:

```
User_Alias      ME=lavander
User_Alias      THEM= pooh, tigger, eeyore, piglet, owl, gopher, kanga, roo
```

- Command Aliases
  This section is used to give a "simpler" name to either a single or group of commands.
  For example,:

```
Cmnd_Alias      DUMPS=/usr/bin/mt, /usr/sbin/dump, /usr/sbin/rdump,\
                      /usr/sbin/restore, /usr/sbin/rrestore
Cmnd_Alias      KILL=/usr/bin/kill
Cmnd_Alias      PRINTING=/usr/sbin/lpc, /usr/bin/lprm
```

- User Privileges
  This is the section where you bind the three alias sections together to create your user rules to determine what a particular user (or group of users) are allowed to run and where.
  Allow root to run everything on all servers:

```
root            ALL = ALL
```

  Allow me (lavander) to run all command aliases on my home machine:

```
ME              HOME = DUMPS, KILL, PRINTING
```

  Allow a group of administrators to run backups on the "kitchen" servers:

```
THEM            KITCHEN = DUMPS
```

You do not need to use the aliases sections in *sudoers*, but it can make things more convenient to extend permissions of one user (or group of users) to another not to mention making the rules

significantly easier to read should you define the aliases with logical names.
In this example we allow the user *bob* to run the reboot command on server mainframe, without using any aliases:

```
bob        mainframe.system.net = /usr/sbin/reboot
```

To use your newly granted *sudo* access, simply prefix the command you want to run as root with the *sudo* command. So, let's assume that you have been granted the ability to change user's passwords through sudo, and today the user *sillyguy* requests their password changed. If you were running the command as root, you would simply run this command as:

```
/usr/bin/passwd sillyguy
```

To run this command under sudo, you would run:

```
sudo /usr/bin/passwd sillyguy
```

You will then be prompted to enter your login password, and assuming that you type in the correct password, the passwd command will proceed as if it were being run from a root prompt.

There is really no security built into *sudo* since it relies on the user's system password for authentication. In the default configuration, once you have authenticated through *sudo,* there is a five minutes grace period within which a user can enter another *sudo*-enabled command without having to supply their password. Each time *sudo* is called, the timestamp is updated and the countdown begins again. Unfortunately, there is no way to disable this behaviour and force password authentication for every *sudo* transaction nor is it easy to change the default timeout value as this is currently a compile-time option.

Despite some of the drawbacks to using *sudo*, we believe that it's advantages out way it's disadvantages and should be considered for deployment on all servers.

Network Time Protocol (NTP)

The importance of accurate time on your systems cannot be stressed enough. For day-to-day operations, having time synchronized is not that important. However, should you need to compare logs from between the web server and the log server (trying to track down some sort of network abnormality, for example) it is important that both servers are logging with the same time. Also, it is imperative that your clocks are synchronized when dealing with incidents involving the law. Time that is not accurate casts doubt on the validity of data and can cause problems when trying to prosecute anyone who is caught attacking or infiltrating your systems.

Network Time Protocol (NTP) will keep all system clocks in synchronization with one another. By tying this system into some of the networked atomic clocks available on the internet, you can get very accurate times on your systems that will be synchronized to the real time. The distance an NTP server is from a reliable time source (like an atomic clock) is called a *stratum* where the atomic clock itself is a Stratum 0 time source, and the NTP server connected to it is called a Stratum 1 time source. An NTP server connected to that one would be considered Stratum 2, since it is 2 "hops" away from the trusted source.

While NTP is distributed with most versions of Solaris, the version is usually significantly far behind the most recent version. The current stable version of NTP is 4.1.0 and is available at

http://www.eecis.udel.edu/~ntp. A list of publicly accessible time servers is available from http://www.eecis.udel.edu/~mills/ntp/servers.htm.

Given the configuration of the GIAC Enterprises network, we would recommend setting up your NTP Servers in a location that is accessible from most of the servers in the network. An obvious candidate that stands out for us are the GIAC DNS Servers, from which all servers get their name resolution. It is important to have more than one NTP server configured within your network, since having a single time source can also lead to problems should the source data become incorrect for any reason. (Disconnection from their NTP source/peer, hacker corrupting the time on your only time source.) Each of the DNS servers should point to three independent outside time sources, for a total of 6 sources. It is likely that your time sources will be slightly out of synchronization so will make an educated guess as to the "real" time. The software turns each time value into a range, by adding & subtracting an error threshold to each time source. Where the majority of the ranges overlap, NTP considers that to be the "real" time. If one of the sources is dramatically out of range, it is ignored completely.

When determining which time sources to use, keep in mind that each server has set different access policies. Insure that you follow any instructions provided upon selection of a particular time service. Once you have arranged for time service, you will need to add these servers to the ntp.conf file. Below, find a sample ntp.conf for the NTP server running on *lion*:

```
server 128.100.103.252      # tick.utoronto.ca
server  209.87.233.53       # clock.chu.nrc.ca
server  142.3.100.15        # timelord.uregina.ca

peer    IP Censored         # tiger.giac-enterprises.com
```

In our example the server will synch time with three Stratum 2 NTP servers, and use the other DNS server as a "sanity check" (known as a peer). This would mean that the 2 NTP Servers running on GIAC's DNS Servers would be considered Stratum 3 sources.

This is fine until access to the Stratum 2's are dropped because of an issue with the PIX firewall, or some other network event causes GIAC Enterprises to loose connections to these NTP servers. When an NTP server looses its connections to it's upstream stratums, it fails into the Stratum 16 configuration, which identifies it as being disconnected. At this point, the NTP clients on the web servers would begin to loose their synchronization as they are no longer talking with the NTP network. In order to deal with this problem, NTP allows you to define pseudo-clocks in ntp.conf which "trick" the NTP server into thinking it still has a connection. Servers who have pseudo-clocks defined do not drop to Stratum 16 when disconnected, instead they will switch to a stratum level that is defined in ntp.conf. An entry like the one below defines a pseudo-clock.

```
server 127.127.1.1
fudge 127.127.1.1   stratum 13
```

The 127.127.1.X network is used to identify the special pseudo-clocks. 127.127.1.1 would be the first instance of the pseudo-clock and 127.127.1.2 would be the second, and so on.

We recommend that you use Stratum 13 as your pseudo-clock since NTP will allow for some fluxuation in the path to your upper time sources. NTP will always take the lowest Stratum signals as priority, so we want to have defined our network clock suitably low.

After the DNS servers are setup, the rest of GIAC's server should be configured to get the time. Using *panther* as our examples, the NTP clients on this would have an ntp.conf file that looks similar to the following:

```
server  IP Censored        # lion.giac-enterprises.com
server  IP Censored        # tiger.giac-enterprises.com

peer    IP Censored        # leopard.giac-enterprises.com
```

The ntp.conf file for *leopard* would be the same, with *panther* configured as the peer. Each server should point at both NTP servers for redundancy and accuracy – for much the same reason we configured 3 Stratum 2 time sources on each of the DNS servers.

Sun has created a number of documents discussing NTP:

Introduction to NTP	http://www.sun.com/blueprints/0701/NTP.pdf
Basic NTP Administration & Architecture	http://www.sun.com/blueprints/0801/NTPpt2.pdf
NTP Monitoring & Troubleshooting	http://www.sun.com/blueprints/0901/NTPpt3.pdf

Password Verification Program (John the Ripper)

In order to insure that users are using "good" passwords, we ran a password verification program (also known as a password cracker) against the passwords on *panther*. The tool we used is called John the Ripper (www.openwall.com/john), which is a freeware application that can check a variety of encrypted passwords. Since both web servers are running Solaris, the passwords have been encrypted with DES, which is one of the supported encryption methods. Documents can be found at this same site that will help with the installation and configuration of the software.

John the Ripper works by encrypting strings and comparing them to the encrypted passwords in the password file of the server. There are two popular modes for running John:

–single	Tries permutations of the user's full name and username.
-wordlist	Tries permutations of a specified file containing a wordlist. This wordlist is usually comprised of a number of online dictionaries with duplicates removed.

The permutation rules are set in the john.ini file, with specific sections for each mode described above. Permutations can be set to try each string in a number of different manners, for example reversing the letters (drowssap) and replacing the letters with numbers and symbols (p@55w0rd).

Aside from the concerns discussed in Access Rules & User Limitations, the passwords currently in use were well quite well built. John was able to determine a small number of passwords after a sustained period but in general, the passwords were difficult to crack. We have left a copy of the permutation rules such that you can run these scans on a regular basis. We do strongly recommend that administrators continue to update the wordlists and perhaps add permutations to enhance the password guessing ability of John the Ripper.

Virus Protection Analysis

Since all of the servers in the environment are UNIX based, the requirement for Virus Protection is limited as most of these parasitic programs target Microsoft Windows-based networks. However, with the number of viruses and worms on the rise, IT managers may want to consider deploying Virus Protection on their UNIX servers so that they are prepared when viruses that do target these servers are unleashed.

A word of caution – since there has not been a need for significant UNIX Virus protection, the software tends not to be as mature as the Windows based versions. For example, as part of the McAffee Anti-Virus Suite, they have a product for Windows called "VShield" that monitors active files. While McAffee does have a UNIX based virus scanner, they do not have an equivalent for "VShield". As a result, the only way to scan for viruses on a UNIX server is to run a full scan against a system, or an individual file system. These scans tend to be quite CPU and disk intensive, so you will not want to run these during peak times and can take a significant amount of time on systems with large filesystems.

Conclusion & Summaries

While the overall security of GIAC Enterprises' network is quite good, there are definitely some areas for improvement. For example, there are servers are running operating systems that have not had security updates performed on them for several months, if not years. Also, some application versions are significantly out of date and some contain documented security vulnerabilities. Web servers used to conduct financial transactions are not doing so in a secure manner and can easily be upgraded to a more secure model. A number of simple changes to the environment can make a significant increase in the reliability and security of the environment at GIAC Enterprises.

Top 10 Violations

1. Servers have not had Operating System patches installed recently
2. Conducting financial transactions through an insecure HTTP communication
3. Application software is several revisions behind current versions.
   Some versions currently being run containing documented security violations.
4. Servers not configured by role – servers are running, or have installed, unnecessary software.
5. Lack of Policy on Response to Major Security Vulnerabilities
6. Current Disaster Recovery Plan outdated and untested.
7. Incorrect or Absent Identification in Restricted Areas
8. Negligible Intrusion Detection Capabilities present.
9. Data in Server Inventory Database is outdated
10. Little to no documentation exists or is available on most corporate policies.

Top 10 Recommendations

1. Apply all Operating System patches
2. Rebuild Secondary Server to upgrade operating system and major applications
3. Audit access rules and user accounts on a regular basis
4. Consider installing more secure versions of common programs (telnet)
5. Consider deploying Secure HTTP for online financial transactions
6. Disable all services not required for the server's current role.
7. Remove all software not being used.
8. Develop & Test Disaster Recovery Plan
9. Keep abreast of current security vulnerabilities in operating system and applications
10. Consider deploying an Intrusion Detection System (IDS) to augment existing security

Details on the violations and recommendations presented here are covered in detail throughout the report. Consult the appropriate section to determine the steps required to implement the recommendation.

Appendix

Patchdiag Output from panther

```
================================================================================
System Name: panther SunOS Vers: 5.6 Arch: sparc
Cross Reference File Date: Dec/10/01

PatchDiag Version: 1.0.4
================================================================================
Report Note:

Recommended patches are considered the most important and highly recommended patches
that avoid the most critical system, user, or security related bugs which have been
reported and fixed to date.  A patch not listed on the recommended list does not imply
that it should not be used if needed.  Some patches listed in this report may have certain
platform specific or application specific dependencies and thus may not be applicable to your
system.  It is important to carefully review the README file of each patch to fully determine
the applicability of any patch with your system.
================================================================================
INSTALLED PATCHES
Patch  Installed Latest   Synopsis
  ID   Revision  Revision
------ --------- -------- ------------------------------------------------------
105160    02        14    CDE 1.2: dtterm libDtTerm.so.1 patch
105181    19        30    SunOS 5.6: Kernel update patch
105189    02        03    OBSOLETED by 106040
105210    27        44    SunOS 5.6: libaio, libc & watchmalloc patch
105214    01     CURRENT  OBSOLETED by 105181
105216    03        04    SunOS 5.6: /usr/sbin/rpcbind patch
105222    03     CURRENT  OBSOLETED by 105181
105223    05     CURRENT  Obsoleted by: 105181-25 SunOS 5.6: pln/soc drivers & ssafirmware p
105284    31        47    Motif 1.2.7: Runtime library patch
105338    25        27    CDE 1.2: dtmail patch
105356    13        18    SunOS 5.6: /kernel/drv/ssd and /kernel/drv/sd patch
105357    04     CURRENT  SunOS 5.6: /kernel/drv/ses patch
105360    10        41    Creator 2.6: FFB Graphics Patch
105361    03        11    VIS/XIL 2.6: Graphics Patch
105362    08        36    PGX 2.6: M64 Graphics Patch
105363    05        38    Elite3D 2.6: AFB Graphics Patch
105364    01        02    SunOS 5.6: SX Graphics Patch
105375    19        26    SunOS 5.6: sf & socal driver patch
105377    03        05    SunOS 5.6: BCP patch
105379    05        07    SunOS 5.6: /kernel/misc/nfssrv patch
105390    02     CURRENT  SunOS 5.6: SGML Manual Pages Patch
105393    07     CURRENT  OBSOLETED by 105621
105395    06        07    SunOS 5.6: /usr/lib/sendmail patch
105397    02     CURRENT  SunOS 5.6: /usr/sbin/passmgmt patch
105400    02     CURRENT  SunOS 5.6: Greek keyboard layout incorrect on Sparc
105401    25        35    SunOS 5.6: libnsl and NIS+ commands patch
105405    01        03    SunOS 5.6: libcurses.a & libcurses.so.1 patch
105407    01     CURRENT  SunOS 5.6: /usr/bin/volrmmount patch
105416    01     CURRENT  SunOS 5.6: /usr/lib/acct/acctdisk patch
105426    01     CURRENT  SunOS 5.6: /usr/lib/libtnfprobe.so.1 patch
105464    02     CURRENT  OpenWindows 3.6: Multiple xterm fixes
105486    02        07    SunOS 5.6: /kernel/fs/hsfs patch
105490    07     CURRENT  OBSOLETED by 107733
105492    02     CURRENT  Obsoleted by: 105181-25 SunOS 5.6: cgsix driver patch
105497    01     CURRENT  OpenWindows 3.6: printtool patch
105516    01        06    SunOS 5.6: /usr/lib/fs/ufs/fsck and mountall patch
105518    01     CURRENT  OBSOLETED by 105395
105528    01     CURRENT  SunOS 5.6: /kernel/drv/be patch
105529    08        11    SunOS 5.6: /kernel/drv/tcp patch
105558    04     CURRENT  CDE 1.2: dtpad patch
105562    03     CURRENT  SunOS 5.6: chkey and keylogin patch
105564    02        04    SunOS 5.6: /kernel/misc/rpcsec patch
105566    08        11    CDE 1.2: calendar manager patch
105568    16        23    SunOS 5.6: /usr/lib/libthread.so.1 patch
105570    01        05    SunVideo 1.3: Patch
105572    03        11    OBSOLETED by 106625
105580    14        18    SunOS 5.6: /kernel/drv/glm patch
105600    15        19    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/isp patch
105604    09     CURRENT  OBSOLETED by 105181
105615    07        08    SunOS 5.6: /usr/lib/nfs/mountd patch
105618    01     CURRENT  OpenWindows 3.6: Xcms patch
105621    19        25    Obsoleted by: 105181-25 SunOS 5.6: c2audit, libbsm and cron patch
105630    01        03    CDE 1.2: libDtWidget patch
105633    36        60    OpenWindows 3.6: Xsun patch
```

```
105642    05        08    SunOS 5.6: prtdiag patch
105651    06        12    Obsoleted by: 105181-25 SunOS 5.6: ac/environ/fhc/sysctrl driver p
105654    03   CURRENT    SunOS 5.6: driver_aliases/driver_classes/name_to_major patch
105665    03   CURRENT    SunOS 5.6: /usr/bin/login patch
105667    02        03    SunOS 5.6: /usr/bin/rdist patch
105669    10        11    CDE 1.2: libDtSvc Patch
105686    02   CURRENT    OBSOLETED by 105621
105693    03        11    SunOS 5.6: cachefs patch
105703    22        27    CDE 1.2: dtlogin patch
105705    02   CURRENT    SunOS 5.6: /usr/kernel/drv/audiocs patch
105718    02        04    SunOS 5.6: /usr/bin/su patch
105720    12        19    SunOS 5.6: /kernel/fs/nfs patch
105722    03        07    SunOS 5.6: /usr/lib/fs/ufs/ufsdump and ufsrestore patch
105724    01   CURRENT    OBSOLETED by 105722
105736    01   CURRENT    OBSOLETED by 105395
105741    06        09    SunOS 5.6: /kernel/drv/ecpp patch
105742    03        05    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/le patch
105743    01   CURRENT    OBSOLETED by 107228
105746    01        03    SunOS 5.6: /usr/bin/cpio patch
105755    07        10    SunOS 5.6: libresolv, in.named, named-xfer, nslookup, nstest patch
105757    01   CURRENT    SunOS 5.6: /usr/bin/echo patch
105778    01   CURRENT    SunOS 5.6: /kernel/fs/specfs patch
105780    04        05    SunOS 5.6: /kernel/fs/fifofs patch
105786    11        14    SunOS 5.6: /kernel/drv/ip driver patch
105792    02        07    SunOS 5.6: /usr/sbin/tar patch
105795    03        08    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/hme patch
105797    06   CURRENT    OBSOLETED by 105356
105798    03   CURRENT    SunOS 5.6: sun4m, sun4u & sun4u1 cprboot patch
105800    06        07    SunOS 5.6: /usr/bin/admintool, y2000 patch
105802    11        16    OpenWindows 3.6: ToolTalk patch
105836    02        03    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/qe patch
105837    03   CURRENT    CDE 1.2: dtappgather Patch, including SDE 1.0 installations
105845    01   CURRENT    OBSOLETED by 105621
105847    01        11    SunOS 5.6: /kernel/drv/st.conf and /kernel/drv/st patch
105867    01   CURRENT    SunOS 5.6: /usr/sbin/tapes patch
105924    03        13    SunOS 5.6: kbd, se and zs drivers patch
105926    01   CURRENT    Obsoleted by: 105792-06 SunOS 5.6: /usr/sbin/static/tar patch
105953    01   CURRENT    SunOS 5.6: /usr/bin/xargs patch
105988    01   CURRENT    SunOS 5.6: /usr/sbin/rwall patch
105990    01        05    SunOS 5.6: vi/ex/edit/view/vedit patch
106025    01   CURRENT    CDE 1.2: sdtfprop patch for group permissions
106027    08        10    CDE 1.2 / SDE 1.0: dtsession patch
106029    01        05    SunOS 5.6: /usr/ccs/bin/sccs and /usr/ccs/bin/make patch
106031    02   CURRENT    OBSOLETED by 105181
106033    01   CURRENT    OBSOLETED by 105621
106035    01   CURRENT    SunOS 5.6: /usr/bin/getopt patch
106040    13        17    SunOS 5.6: X Input & Output Method patch
106044    01        03    Obsoleted by: 105210-38 SunOS 5.6: /usr/lib/nss_nisplus.so.1 patch
106049    01        03    SunOS 5.6: /usr/sbin/in.telnetd patch
106075    01   CURRENT    OBSOLETED by 105621
106084    01        04    OBSOLETED by 107013
106112    05        06    CDE 1.2: dtfile patch
106123    04        05    SunOS 5.6: sgml patch
106125    09        12    SunOS 5.6: Patch for patchadd and patchrm
106138    01   CURRENT    OpenWindows 3.6: mp fails to set correct A4 paper size information
106141    01   CURRENT    SunOS 5.6: /usr/bin/mkdir patch
106168    02   CURRENT    Obsoleted by: 105181-25 SunOS 5.6: dma driver patch
106169    02   CURRENT    Obsoleted by: 105181-25 SunOS 5.6: sbusmem driver patch
106170    02        03    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/esp patch
106171    01   CURRENT    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/lebuffer patch
106172    02        05    Obsoleted by: 105181-25 SunOS 5.6: /kernel/drv/fas patch
106173    02        03    Obsoleted by: 105181-25 SunOS 5.6: /kernel/misc/scsi patch
106183    03        05    SunOS 5.6: cfgadm utility & libraries
106193    03        06    SunOS 5.6: Patch for Taiwan timezone
106216    01        03    SunOS 5.6: /platform/sun4u/kernel/drv/envctrl patch
106219    01        03    SunOS 5.6: luxadm.1m Manual Page Patch
106222    01   CURRENT    OpenWindows 3.6: filemgr (ff.core) fixes
106226    01        02    SunOS 5.6: /usr/sbin/format patch
106235    04        09    SunOS 5.6: lp patch
106242    02        03    CDE 1.2: libDtHelp.so.1 fixes
106257    05   CURRENT    SunOS 5.6: /usr/lib/libpam.so.1 patch
106260    01   CURRENT    SunOS 5.6: Manual Pages Patch for ffbconfig.1m
106261    01   CURRENT    SunOS 5.6: Manual Pages Patch cfgadm.1m config_admin.3x libcfgadm.
106262    01   CURRENT    SunOS 5.6: Manual Pages Patch for qfe.7d
106271    06        08    SunOS 5.6: /usr/lib/security/pam_unix.so.1 patch
106301    01        04    SunOS 5.6: /usr/sbin/in.ftpd patch
106317    01   CURRENT    SunOS 5.6:  upgrade_script terminated abnormally during upgrade
106323    01   CURRENT    SunOS 5.6: /etc/inet/services patch
106415    03        04    OpenWindows 3.6: xdm patch
106437    03   CURRENT    CDE 1.2: Print Manager Patch
106439    06        09    SunOS 5.6: /usr/sbin/syslogd patch
```

```
106448    01     CURRENT  SunOS 5.6: /usr/sbin/ping patch
106495    01     CURRENT  SunOS 5.6: truss & truss support library patch
106522    03        04    SunOS 5.6: /usr/bin/ftp patch
106569    01     CURRENT  SunOS 5.6: libauth.a & libauth.so.1 patch
106592    02        04    SunOS 5.6: /usr/lib/nfs/statd patch
106625    05        13    SunOS 5.6: libsec.a, libsec.so.1 and /kernel/fs/ufs patch
106648    01     CURRENT  OpenWindows 3.6: libce suid/sgid security fix
106649    01     CURRENT  OpenWindows 3.6: libdeskset patch
106650    04     CURRENT  OpenWindows 3.6: mailtool attachment security patch
106828    01     CURRENT  SunOS 5.6: /usr/bin/date patch
106834    01        02    SunOS 5.6: cp/ln/mv patch
107336    01     CURRENT  OpenWindows 3.6: KCMS configure tool has a security vulnerability
107434    01     CURRENT  CDE 1.2: Spell checking occasionally kills mail
107492    01     CURRENT  SunOS 5.6: Y2000, runacct cannot update /var/adm/acct/sum/loginlog
107565    02        03    SunOS 5.6: /usr/sbin/in.tftpd patch
107618    01        02    SunOS 5.6: patch /usr/sbin/vold
107758    01     CURRENT  SunOS 5.6: Pax incorrectly change mode of symlink target file
107766    01     CURRENT  SunOS 5.6: ASET cklist reports unchanged 6month older files as new
107774    01     CURRENT  SunOS 5.6: inetd denial-of-service attack
107988    01     CURRENT  SunOS 5.6: Patch for SPARCompiler Binary Compatibility Libraries
107991    01        02    SunOS 5.6: /usr/sbin/static/rcp patch
108199    01     CURRENT  CDE 1.2: dtspcd Patch
108201    01     CURRENT  CDE 1.2: dtaction Patch
108492    01     CURRENT  SunOS 5.6: Snoop may be exploited to gain root access
108660    01     CURRENT  SunOS 5.6: Patch for sadmind
================================================================================

UNINSTALLED RECOMMENDED PATCHES

Patch  Ins Lat Age  Require    Incomp  Synopsis
  ID   Rev Rev        ID         ID
------ --- --- --- --------- --------- ----------------------------------------
105403 N/A  04 175                     SunOS 5.6: ypbind/ypserv patch
105472 N/A  08 166                     SunOS 5.6: /usr/lib/autofs/automountd patch
105552 N/A  03 606                     SunOS 5.6: /usr/sbin/rpc.nisd_resolv patch
106285 N/A  03 357                     SunOS 5.6: /kernel/sys/msgsys patch
106292 N/A  11 364                     SunOS 5.6: pkgadd/pkginstall & related utilities
106303 N/A  03 125 106271-07           SunOS 5.6: /usr/lib/netsvc/yp/rpc.yppasswdd patch
106361 N/A  13  28                     SunOS 5.6: csh/jsh/ksh/rksh/rsh/sh patch
106429 N/A  02 306                     SunOS 5.6: /kernel/drv/mm patch
106468 N/A  05  50                     SunOS 5.6: /usr/bin/cu and usr/bin/uustat patch
106639 N/A  06 133                     SunOS 5.6: /kernel/strmod/rpcmod patch
106882 N/A  02 470                     SunOS 5.6: /usr/lib/nfs/nfsd patch
106894 N/A  01 1072                     Obsoleted by: 106468-04 SunOS 5.6: /usr/bin/uux patch
107298 N/A  03  57                     SunOS 5.6: ntpdate and xntpd patch
107326 N/A  02  78                     SunOS 5.6: rlmod and telmod patch
107490 N/A  01 981                     SunOS 5.6: savecore doesn't work if swap slice is over 2G
107733 N/A  09 433                     SunOS 5.6: Linker patch
108307 N/A  02 606                     SunOS 5.6: keyserv fixes
108333 N/A  02 482                     SunOS 5.6: jserver buffer overflow
108346 N/A  03 606                     SunOS 5.6: patch usr/sbin/rpc.nispasswdd
108468 N/A  02 565                     SunOS 5.6: ldterm streams module fixes
108499 N/A  01 687                     SunOS 5.6: ASET sets the gid on /tmp, /var/tmp when
setting med hi
108804 N/A  02 175                     SunOS 5.6: /usr/bin/tip patch
108890 N/A  01 606                     SunOS 5.6: patch /usr/lib/netsvc/yp/ypxfrd
108893 N/A  01 606                     SunOS 5.6: patch /usr/lib/netsvc/yp/rpc.ypupdated
108895 N/A  01 606                     SunOS 5.6: patch /usr/sbin/rpc.bootparamd
109266 N/A  02  97                     SunOS 5.6: /usr/bin/mail has buffer overflow
109339 N/A  02 287                     SunOS 5.6: nscd's size grows - TTL values not implemented
109388 N/A  01 557                     SunOS 5.6: patch /usr/vmsys/bin/chkperm
109719 N/A  01 340                     SunOS 5.6: arp should lose set-gid bid
110990 N/A  01 208                     SunOS 5.6: Patch for ttymon
111029 N/A  01 243                     SunOS 5.6: /kernel/sys/semsys patch
111039 N/A  02  68                     SunOS 5.6: /usr/bin/bdiff and /usr/bin/sdiff patch
111109 N/A  01 216                     SunOS 5.6: Patch to /usr/bin/nawk
111236 N/A  01 214                     SunOS 5.6: Patch for /usr/sbin/in.fingerd
111240 N/A  01 214                     SunOS 5.6: Patch to /usr/bin/finger
111560 N/A  01 175                     SunOS 5.6: dmesg security problem
111572 N/A  01 166                     SunOS 5.6: ar_open failure can lead to stale queue &
memory corrup
111664 N/A  01 165                     SunOS 5.6: bzip patch
111859 N/A  01 106                     SunOS 5.6: Buffer overflow in whodo via $TZ
112073 N/A  02  35                     SunOS 5.6: /usr/bin/mailx security problem
================================================================================

UNINSTALLED SECURITY PATCHES

NOTE: This list includes the Security patches that are also Recommended

Patch  Ins Lat Age  Require    Incomp  Synopsis
```

```
  ID   Rev Rev        ID          ID
------ --- --- --- --------- --------- ----------------------------------------
105403 N/A  04 175                     SunOS 5.6: ypbind/ypserv patch
105552 N/A  03 606                     SunOS 5.6: /usr/sbin/rpc.nisd_resolv patch
106303 N/A  03 125 106271-07           SunOS 5.6: /usr/lib/netsvc/yp/rpc.yppasswdd patch
106361 N/A  13  28                     SunOS 5.6: csh/jsh/ksh/rksh/rsh/sh patch
106468 N/A  05  50                     SunOS 5.6: /usr/bin/cu and usr/bin/uustat patch
106629 N/A  23 354           105181-08 SunOS 5.6: CS6400 kernel update patch
106639 N/A  06 133                     SunOS 5.6: /kernel/strmod/rpcmod patch
106882 N/A  02 470                     SunOS 5.6: /usr/lib/nfs/nfsd patch
106894 N/A  01 1072                     Obsoleted by: 106468-04 SunOS 5.6: /usr/bin/uux patch
107298 N/A  03  57                     SunOS 5.6: ntpdate and xntpd patch
107326 N/A  02  78                     SunOS 5.6: rlmod and telmod patch
107733 N/A  09 433                     SunOS 5.6: Linker patch
108307 N/A  02 606                     SunOS 5.6: keyserv fixes
108333 N/A  02 482                     SunOS 5.6: jserver buffer overflow
108346 N/A  03 606                     SunOS 5.6: patch usr/sbin/rpc.nispasswdd
108468 N/A  02 565                     SunOS 5.6: ldterm streams module fixes
108499 N/A  01 687                     SunOS 5.6: ASET sets the gid on /tmp, /var/tmp when
setting med hi
108804 N/A  02 175                     SunOS 5.6: /usr/bin/tip patch
108890 N/A  01 606                     SunOS 5.6: patch /usr/lib/netsvc/yp/ypxfrd
108893 N/A  01 606                     SunOS 5.6: patch /usr/lib/netsvc/yp/rpc.ypupdated
108895 N/A  01 606                     SunOS 5.6: patch /usr/sbin/rpc.bootparamd
109100 N/A  02  28                     SunOS 5.6: patch usr/sbin/mkdevmaps
109266 N/A  02  97                     SunOS 5.6: /usr/bin/mail has buffer overflow
109339 N/A  02 287                     SunOS 5.6: nscd's size grows - TTL values not implemented
109388 N/A  01 557                     SunOS 5.6: patch /usr/vmsys/bin/chkperm
109719 N/A  01 340                     SunOS 5.6: arp should lose set-gid bid
110883 N/A  01 130                     SunOS 5.6: useradd date format fixes
111039 N/A  02  68                     SunOS 5.6: /usr/bin/bdiff and /usr/bin/sdiff patch
111236 N/A  01 214                     SunOS 5.6: Patch for /usr/sbin/in.fingerd
111240 N/A  01 214                     SunOS 5.6: Patch to /usr/bin/finger
111560 N/A  01 175                     SunOS 5.6: dmesg security problem
111645 N/A  01 127                     SunOS 5.6:  BCP libmle buffer overflow
111859 N/A  01 106                     SunOS 5.6: Buffer overflow in whodo via $TZ
112073 N/A  02  35                     SunOS 5.6: /usr/bin/mailx security problem
================================================================================

UNINSTALLED Y2K PATCHES

NOTE: This list includes the Y2K patches that are also Recommended

Patch  Ins Lat Age  Require    Incomp  Synopsis
  ID   Rev Rev        ID          ID
------ --- --- --- --------- --------- ----------------------------------------
108667 N/A  03 671                     CDE 1.2: perfmeter is not Y2K compliant in SunOS 5.6
Supplement
108671 N/A  03 518                     OpenWindows 3.6: Calendar Manager patch
================================================================================
```

NMAP Results

Nmap (V. nmap) scan initiated 2.53 as: nmap -sS -O -v -T polite -oN /home/lavander/nmapscan.txt *IP Censored*

Interesting ports on www.giac-enterprises.com (*IP Censored*):
(The 1517 ports scanned but not shown below are in state: closed)

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
80/tcp	open	http

TCP Sequence Prediction: Class=truly random
Difficulty=9999999 (Good luck!)
Sequence numbers: AD294CBB D6006D9E 86188F6C 756E2F5F 17657D2D 23CA8313

No OS matches for host (If you know what OS is running on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).

TCP/IP fingerprint:
TSeq(Class=TR)
T1(Resp=Y%DF=Y%W=2297%ACK=S++%Flags=AS%Ops=NNTNWME)
T2(Resp=N)
T3(Resp=N)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=N)
PU(Resp=N)
# Nmap run completed at Tue Nov 27 15:26:19 2001 -- 1 IP address (1 host up) scanned in 665 seconds

Nessus Results

List of open ports :

- *ftp (21/tcp) (Security notes found)*
- *telnet (23/tcp) (Security warnings found)*
- *http (80/tcp) (Security notes found)*
- *general/udp (Security notes found)*
- *general/tcp (Security notes found)*

Information found on port ftp (21/tcp)

Remote FTP server banner :
220-##
220-
220- This system is restricted to authorized personnel. Only authorized
220- work is to be done. Use of this system is an agreement to monitoring.
220- Any unauthorized use is subject to disciplinary action.
220-
220-Access Denied for *IP Censored*.
220-
220-##

Warning found on port telnet (23/tcp)

The Telnet service is running.
This service is dangerous in the sense that it is not ciphered - that is, everyone can sniff the data that passes between the telnet client and the telnet server. This includes logins and passwords.

You should disable this service and use OpenSSH instead. (www.openssh.com)

Solution : Comment out the 'telnet' line in /etc/inetd.conf.

Risk factor : Low
CVE : CAN-1999-0619

Information found on port telnet (23/tcp)

Remote telnet banner :
##

This system is restricted to authorized personnel. Only authorized
work is to be done. Use of this system is an agreement to monitoring.
Any unauthorized use is subject to disciplinary action.

Access Denied for *IP Censored*

##

Information found on port http (80/tcp)

The remote web server type is :

–

Information found on port general/udp

For your information, here is the traceroute to *IP Censored* - ?

Information found on port general/tcp

QueSO has found out that the remote host OS is
* Solaris 2.x

CVE : CAN-1999-0454

Useful Websites

Tools/Applications Mentioned

http://www.insecure.org	NMAP Port Scanner. Also great "hacking" information
http://www.nessus.org	Nessus Vulnerability Scanner
http://www.courtesan.com/sudo	SUDO
http://www.ssh.fi	SSH
http://www.openssh.com	OpenSSH (Free Alternative to SSH)
http://www.wu-ftpd.org	WU-FTP
ftp://ftp.porcupine.org/pub/security/index.html#software	TCP Wrappers Also contains some security papers.
http://www.openwall.com/john	John the Ripper (Password Cracker)
http://www.tripwire.com	Tripwire
http://www.apache.org	Apache Web server

Vulnerability Mailing Lists/Web Sites

http://www.sans.org	**SANS Institute**
– http://www.sans.org/infosecFAQ/index.htm	Information Security Reading Room
– http://www.sans.org/top20.htm	Top 20 Internet Security Vulnerabilities
http://www.cert.org	**CERT Co-ordination Centre**
– http://www.cert.org/nav/index_red.html	Vulnerabilities, Incidents & Fixes
– http://www.cert.org/nav/index_green.html	Security Practices & Evaluations
http://xforce.iss.net	Internet Security Systems, Inc: X-Force
http://www.nmrc.org/faqs/www/index.html	Web Hacking FAQ

Other Websites

http://www.sun.com	**Sun Microsystems**
– http://docs.sun.com	Sun Documentation Website
– http://sunsolve.sun.com	SunSolve Website (Solaris Patches, InfoDocs, etc.)
http://ars.userfriendly.org/cartoons/?id=19971127	Example of poor disaster recovery

References

Apache, The Definitive Guide, Second Edition
Laurie, Ben and Laurie, Peter; O'Reilly & Associates, Inc., February 1999

Essential System Administration, Second Edition
Frisch, Æleen; O'Reilly & Associates Inc., 1995

Practical Unix & Internet Security, Second Edition
Garfinkel, Simson and Spafford, Gene; O'Reilly & Associates, Inc., April 1996

Solaris Operating Environment Security
Noordergraaf, Alex and Watson, Keith; Sun Blue Prints Online, January 2000

Solaris Security Step-By-Step, Version 2.0
Pomeranz, Hal; The SANS Institute, 2001

Track 6 – Securing UNIX: 6.1 Common Issues and Vulnerabilities in UNIX Security
Pomeranz, Hal; The SANS Institute, Tuesday, August 14, 2001

Track 6 – Securing UNIX: 6.2 UNIX Security Tools and Their Uses, 6.2
Pomeranz, Hal; The SANS Institute, Wednesday, August 15, 2001

Track 6 – Securing UNIX: 6.3 Topics in UNIX Security
Pomeranz, Hal; The SANS Institute, Thursday, August 16, 2001

Track 6 – Securing UNIX: 6.4 Running UNIX Applications Securely
Pomeranz, Hal; The SANS Institute, Friday, August 17, 2001

Track 6 – Securing UNIX: 6.5 UNIX Practicum
Cole, Eric and Pomeranz, Hal; The SANS Institute, Saturday, August 18, 2001

Web Security & Commerce
Garfinkel, Simson and Spafford, Gene; O'Reilly & Associates, Inc., June 1997