Security Audit of the Order/Inventory System (OIS)
Submitted by John Notsch
11 Aug 2000

Executive Summary

Prior to making a locally developed Order/Inventory System (OIS) accessible for Internet use, an audit was conducted to 1) determine OIS vulnerability to information security threats and 2) to recommend methods to minimize system vulnerabilities where identified.

Audit Conclusion: The OIS as currently configured is extremely insecure and has numerous vulnerabilities. Minimal system monitoring has been done and system may already be compromised. The OIS should **NOT** be made accessible to users over the Internet until all of the issues listed below are resolved and recommendations implemented.

Identified Problems:
- Servers where designed and implemented for in-house use, little effort was made to properly secure servers.
- Applications and system software are not up to date.
- Configuration problems exist in essential systems files.
- Servers are vulnerable to a wide range of common Internet attacks: password sniffing/cracking, RPC attacks, Web server compromises, denial of service attacks, etc.

Recommended Actions to Resolve Problems:
- Fix operating system vulnerabilities immediately using instructions given to secure in-house operation at a minimum.
- At earliest opportunity, re-install operating systems using methods sufficient to insure a clean, secure, un-compromised environment. Disable all unnecessary services and applications and re-implement operating system fixes.
- Install and configure TCP Wrappers, Secure Shell, and passwd+ to limit server access to known approved users and provide greater assurances of user account security.
- Install Apache Web server with SSL component to provide secure web server.
- Install rpcbind replacement and consider hardware upgrades to minimize RPC vulnerabilities.
- Hold technical discussions and/or negotiations with vendor of client server software to develop a method of securing remote client access to servers.

I. Background

The ABC Company currently operates an Order/Inventory System (OIS) to track the engineering and distribution of custom designed widgets for use with its products. The information contained in the OIS is considered commercially discreet and must be safeguarded from competitors. The OIS was originally designed to operate as an in-house information system with access restricted to company personnel only. System design and implementation followed the idea that system contents should be fully accessible to users. Any required security would come as a result of the operations of existing corporate firewalls and other network security features limiting outside access. As implemented, the OIS is comprised of two servers residing on a local area network with neither server having direct access to the Internet.

The company CEO has decided that business operations will be enhanced by allowing company contractors and business partners direct access to the OIS over the Internet. As a result of this directive, the OIS configuration must be audited and all necessary security precautions must be implemented to provide an acceptable level of security. This document details the results of the audit conducted and makes recommendations on how OIS security can be strengthened to withstand anticipated Internet security threats.

II. Order/Inventory System Configuration and Operation

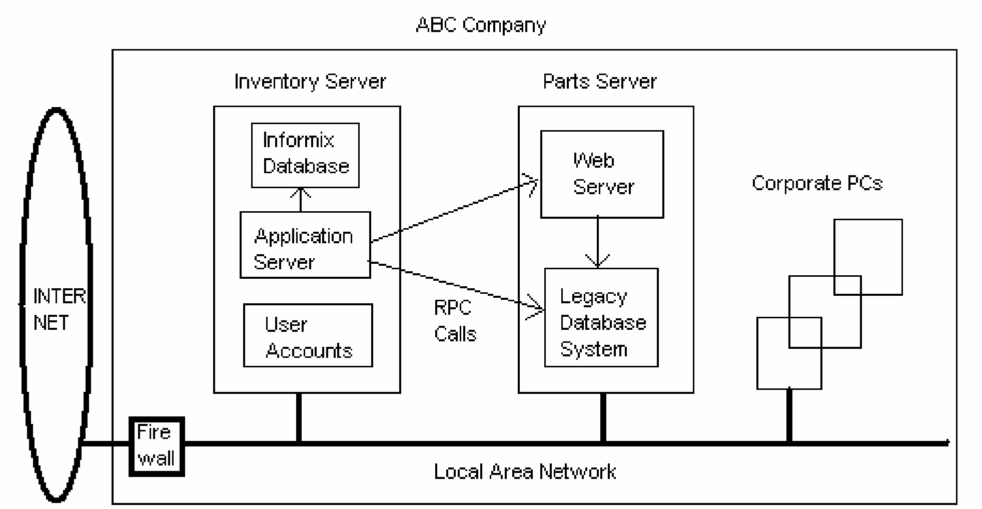

Figure 1.

Figure 1 shows a schematic of the ABC Company network and OIS servers. The OIS is comprised of two Sun workstations running the Solaris 7 operating system. End users interact with the OIS from their desktop computer in one of two ways. The primary interface is through a custom Client/Server application, the only method which allows direct access to corporate data. The secondary interface is through standard telnet/ftp sessions and is limited to the retrieval of reports generated by the Client/Server application or changing user passwords. The key components of the OIS are:

Inventory Server:

- Informix Database - contains all inventory records
- Applications Server - Custom Server Application developed using commercial software
- User accounts - each end-user has an account on this system
- FTP server (not shown) - used for transfer of custom reports generated while on-line

Parts Server:

- Legacy Database System - contains images plus all information that describes a widget.
- Web Server - allows a user to view all information for and an image of a widget
- X-Windows server (not shown) - interface for maintenance of legacy database
- Admin accounts only - non-privileged users not permitted direct access to this system

Corporate PCs:

- Custom client application - primary interface to OIS
- Web browser and telnet/ftp client

Corporate Policy:

- The OIS is a 'read-only' system. End-users can view all information in the system but are not permitted to modify any data contained in the OIS. Only pre-defined 'administrative' accounts are granted privileges to modify OIS contents.
- Prior to gaining access to the OIS, all users must sign a legally binding non-disclosure form stating that they have been informed of corporate policies and that they will not disclose any information obtained from the OIS to competitors. Once users have signed the non-disclosure form, they are considered 'trusted' and are permitted access to all information in the OIS without restriction. Client applications are released to the user only after the forms are signed. Applications are tailored to individual PC environment.
- OIS is managed separately from the corporate administrative and personnel systems. Not all employees require access to the OIS and OIS access is generally available to only a few groups within the company. Fortunately, this separation eliminates the need to include the OIS as part of an network file system or any corporate-wide distributed database configuration. As a result, NFS and NIS/NIS+ are not required on the OIS.

Operation of Client/Server Applications:

Applications using a client/server model are the primary interface for users to the OIS. On the client side, a small client application is installed on each user PC when they are assigned an initial password. When the user starts the client, they are presented with a login box where they are expected to enter their account name and password. After authentication, they are presented with a graphical user interface from which they can query the OIS (both simple and complex queries), display current inventory information, make requests for items in the inventory, and generate a number of various reports. For display of image data, the client uses a local browser application (Netscape, Explorer) already installed on the user's PC. The only reasons a user should ever need to log into the OIS server directly is to either retrieve a report or to change a password.

On the Inventory server, the server application daemon listens for connect requests from clients on a high numbered port and manages the login process. After a successful login, the server daemon starts execution of a locally developed application to paint the interface screen for the user and manages all links to the databases. Queries to the Informix database for inventory data use embedded SQL statements to retrieve data. Queries to the parts database use remote procedure calls to the Parts Server to search for and return the requested data. For viewing of part images and data, the server daemon again uses a remote procedure call to invoke CGI applications running against the Web server on the Parts system. When the user generates reports, the output is directed to either the user's PC or to the their account on the server depending on report type.

III. Auditing Methods Used

To determine the extent of the Order/Inventory System's vulnerabilities, three primary areas of operation were investigated. First, several freely available Unix security tools were employed to test server configurations. Second, OIS administrative personnel were interviewed to determine how the servers were actually operated. Finally, corporate policies and standard operating procedures were reviewed to determine how closely normal operations adhered to expected practices. The security tools chosen for use in auditing were a system integrity scanner, a network scanning tool, and a password cracker. Installation and operation of the system integrity scanner and password cracker were done with the direct approval of, and supervision by, the OIS systems administrator. Network scanning was conducted with the prior written approval of the corporate IS Department. All results obtained from the use of these applications are subject to review by OIS administrative personnel.

Both system servers were originally configured with identical operating system software and settings. Any maintenance, upgrades or patches were installed on both systems in parallel. In discussing system security vulnerabilities, all discussions apply to both servers unless otherwise noted. Any major differences between servers are due to the installation of 3rd party software applications and are noted where appropriate.

Applications Used:

- Tiger version 2.2.4 by Douglass Lee Schales, Dave Hess, Khalid Warraich, Dave Safford - System Integrity and Vulnerability scanner was used to identify potential problems in system configuration.

- nmap version 2.53 by fyodor@dhp.com - Network scanning application used to look for open/vulnerable ports on hosts.

- crack version 5.0 by Alec Muffet - Utility to guess passwords

IV. Order/Inventory System Findings and Vulnerabilities

As in-house, open systems, the two servers comprising the Orders/Inventory System were implemented with little or no thought to security considerations. As shown by a system audit, few measures have been taken to secure either server. Both servers have multiple vulnerabilities ranging from minor problems to major security flaws requiring resolution prior to exposing the servers to the Internet.

A. Operating System Vulnerabilities

The following operating system vulnerabilities were uncovered primarily as a result of a system integrity scan using Tiger:

1. root
   - root logins not restricted to console.
   - Directories in root path not owned by root
2. passwd file
   - Multiple entries for UID=0
   - Two accounts do not have passwords
   - Logins to multiple accounts are disabled but accounts still have valid shell
3. filesystems
   - setuid programs with known vulnerabilities
   - System files have group write permissions
   - System files not owned by root
4. core files
   - No restrictions on the generation of core files on either server.
   - Stack area not protected from user caused buffer overflows.

B. Configuration Vulnerabilities

1. PROM monitor is not in secure mode. VULNERABLE: System does not require password to issue EEPROM commands and may result in denial of service attacks.
2. System Hardware
   - No hardware redundancy in implemented. One server is two years old and the second server is almost four years old. VULNERABLE: If either CPU or disk storage units fail, the OIS will be off-line.
   - RAID hard disk storage units are in use for data storage. All critical data is mirrored to minimize loss of data due to hard disk crashes.
   - Hardware maintenance contracts provide only response within 24 hour on-site hardware support and on-line tech support during normal business hours. VULNERABLE: Potential for extended system downtime for some hardware failures possible.

3. System Services

Operating systems were installed with the Sun Solaris Developer cluster option.

- Systems have sendmail daemons running but do not function as mail servers
- X-Windows server installed but only used on Parts Server
- Applications development tools installed but only used on Parts Server
- Small services (echo, chargen, etc.) installed but not used
- RPC services installed but NIS/NIS+ and NFS not in use.

VULNERABLE: Unnecessary services are installed and operational.

4. Logging

- Only default logging initialized by Solaris install is run. Even these logs are not checked routinely. VULNERABLE: Attacks and intrusions cannot be detected due to lack of adequate logging.

5. Network / Communications

- Servers are not used for communications to Internet. Anonymous FTP is disabled
- Static routes are used to known internal hosts. Use of DNS is minimal.

C. Risks from Third Party Software

1. Databases

- Informix database implemented with appropriate security measures. Access to data is managed at the database connect and table levels, read/write restrictions are used with all database tables. Database administrator account is protected and account passwords are properly restricted. All transactions are logged. Databases are mirrored and backed up. Database files and applications installed and maintained with proper file ownership and access restrictions.
- Parts database implemented with appropriate security measures. Database files and applications are owned by an admin account. The account is protected and admin passwords are properly restricted. End users are granted access permission sufficient only to read.

2. Client/Server Applications

- User logins to the OIS is done through a client application running on the PC. Account names and passwords are not encrypted by the client before transmission. VULNERABLE: Username / passwords are subject to being intercepted by network sniffing tools.
- Inventory Server uses remote procedure calls to interact with Parts Server on remote system. VULNERABLE: Systems subject to common RPC attacks.

3. Web server

- Default installation of a relatively old and insecure web server, NCSA version 1.5. VULNERABLE: Web server subject to attack.
- Web server uses many locally written CGI programs to provide access to Parts database. CGI programs were written in Perl by inexperienced programmers

and may contain flaws. Perl scripts are not protected. VULNERABLE: Unsecured CGI programs may provide easy access to intruders.

D. Administrative Practices and Policies

1. User Accounts

- User accounts are set to a forced password expiration interval of 120 days but password aging is not enforced. Accounts must have passwords but no restrictions are placed on user selected passwords. VULNERABLE: Users may choose and hold easily guessed passwords.
- Passwords were easily guessed by the crack application. More than one half of the passwords tested were guessed by the crack application. VULNERABLE: Sniffed passwords will provide an intruder easy access to the system.
- There are no restrictions on the amount of data that can be stored in a user account (there is a legitimate need for generation of some large reports). VULNERABLE: There is a possibility that a user may fill all available disk space causing a denial of service.

2. System Upgrades

- There are no procedures or policies in place to install software patches and upgrades. Patches are only installed in response to directives from the Corporate IS department. VULNERABLE: Server may be open to attack against known software bugs.

3. System Backups

- All backups are done to DAT tape drives. Full data backups are run in batch mode on Friday nights. Most recent set of backup tapes are stored in a fire-resistant media safe in the corporate office. The second most recent backup tape set is stored off-site at a secured contractor's location 10 miles distant from the corporate office. Backup tapes are transported by a bonded contract courier. Five backup tape sets are in the backup rotation. The oldest retrievable data from backups is up to 34 days old. Program development directories are included in weekly backup. Semi-annually, a complete off-line backup of both servers, data and programs, is run with backup tapes stored in the media safe.
- Informix database is fully backed up once a week on Friday evenings. Incremental backups are run each evening after normal duty hours. Database logging is in effect to log all transactions to database. Legacy database is fully backed up every evening after normal duty hours. Friday backup is included in off-site backup rotation.

4. Physical Security

- OIS servers are located in a secured facility. ID badges are required for entry into the building, all entrances are monitored by video camera, and security personnel are on duty 24 hours a day, seven days a week.
- Servers are located in a secured area with only one entrance, a secured wooden door with a standard lock. There are no drop ceilings or other points of access to the computer room. To gain access to the computer room, persons must pass through an office area staffed by administrative personnel during normal

business hours. VULNERABLE: Wooden door and lock to computer room may be easily compromised.

- Power is supplied by redundant un-interruptable power supplies adequate to power servers for over an hour at full load. Power supplies take their power from a building-wide emergency power grid that activates within seconds of a power outage.
- Network connections are pre-installed into all offices and work areas. All cables are routed through building interstitial spaces to centrally located communications rooms containing routers and links to the main corporate local area network. VULNERABLE: Network is subject to unauthorized internal access or tampering from unattended ports.

5. Network communications
   - The corporate IS department is responsible for configuring and maintaining the firewall between the local area network and the Internet. In general, there is little interaction between corporate IS and the group maintaining the OIS systems. Also, the overwhelming majority of desktop systems and servers elsewhere in the company are Windows-based, not Unix. Because of this, the IS group has little awareness or experience with security requirements for Unix systems. VULNERABLE: OIS administrators have little assurance that corporate firewall offers proper protections for systems.

V. Prioritized List of Vulnerabilities

Three categories of system vulnerabilities are listed. Problems in the first category can be resolved by system reconfiguration that does not require any major system downtime or additional costs incurred. The second category contains vulnerabilities that can be resolved if the organization is willing to accept disruptions to operations in terms of major system reconfiguration or system downtime. This category also includes problems that may require additional hardware costs or investment in programming resources. The third category contains vulnerabilities that can only be resolved with the help of software vendors, problems of low priority that pose little threat, or problems that may be unresolvable at the present time. Within each category, vulnerabilities are listed by priority. Different aspects of the same vulnerability may be listed in more than one category.

A. Vulnerabilities that can be resolved immediately with minimal system disruption:

- root login not restricted to console
- Unnecessary services are installed and operational.
- Not current on OS patches, server may be open to attack against known software bugs.
- Errors in passwd file may allow unauthorized access to server.
- setuid/setgid programs and files with known vulnerabilities.
- System files not owned by root.
- File systems are group writeable.

- Attacks and intrusions cannot be detected due to lack of adequate logging.
- Core files generated on servers
- Buffer overflows caused by user programs possible.

B. Vulnerabilities requiring additional costs, significant system reconfiguration and/or system downtime:

- Several user authentication and password issues must be addressed:
  1) username/passwords transmitted over the network in the clear,
  2) easily guessed passwords and
  3) restricting user access to known IP addresses.
- Web server application and locally developed CGI applications are not secure.
- System subject to common RPC attacks.
- No user disk quotas resulting in potential denial of service if volumes fill to capacity.
- No hardware redundancy, multiple points of failure could cause system downtime.

C. Vulnerabilities that are unresolvable, require modifications to vendor applications, or are considered low priority:

- Passwords in client/server application vulnerable to network sniffing.
- OIS administrators have no control over corporate firewall configuration.
- Computer room door is insecure.
- Vendor technical support available only during normal business hours.
- Network cabling is not secured from unauthorized access or tampering.
- Network communications equipment is not protected by backup power supplies

VI. Recommended Fixes to Identified Vulnerabilities

This section recommends solutions to the server vulnerabilities identified above. In addition to specific instructions, general solutions are also offered. Please note that all recommendations listed in section A make one large assumption; the servers have not yet been compromised by attackers. Although, the system integrity scan by Tiger showed no signs of known intrusion, the servers are insecure and no logging was ever done to detect intrusions. Either one or both of the servers may in fact be infected, there is no way to be certain. The steps in section A will minimize the effects of any future attacks until the time when systems administrators can reinstall the Solaris operating system and configure server security properly. All of the steps mentioned here should also be included as part of the reinstallation process.

A. Implement these instructions immediately to resolve vulnerabilities that can be fixed with minimal system disruption:

Restrict direct root logins to console only

- To prevent remote root logins and provide more logging, restrict direct root logins only to the console. If not at the console, a person must log into a valid user account and then su to the root account. Add the following line to */etc/default/login*:

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

Disable unused system services (Refer to output of nmap in Appendix A)

- These services can be safely removed from both servers:
  - finger - system information need not be provided to remote hosts
  - tftp - ftp should only be executed by valid users to their own account
  - comsat - users unlikely to get mail so no notification is needed
  - echo, chargen, daytime, time, discard
  - unneeded RPC : ttdbserverd, cmsd, rstatd, kcms_server, cachefsd, rusersd

If any of these services are running, use the kill command to stop execution. Until tcp_wrappers can be installed, the */etc/inetd.conf* file should be edited to remove references to these services so they will not be restarted.

- NFS is not used on either server. Use the kill command to stop execution of any running nfs processes. Run the following commands to stop NFS from being restarted on a system reboot:

```
cd /etc/rc2.d
for file in K60nfs.server S73nfs.client S74autofs *cache*
do
        mv $file .NO$file
done
```

- Sendmail is not used on either server. Use the kill command to stop execution of any running sendmail processes. Run the following commands to stop the sendmail daemon from being restarted on a system reboot:

```
cd /etc/rc2.d
mv S88sendmail .NOS88sendmail
mv S80PRESERVE .NOS80PRESERVE
```

Install current operating system and security patches for Solaris 7

- Download latest recommended Solaris 7 patch cluster and security patches from Sun's web site using appropriate methods. Information and patches can be obtained from:

http://sunsolve.Sun.COM/pub-cgi/show.pl

- To install the Solaris 7 patch cluster, download 7_Recommended.zip to the */var/tmp* directory and run commands:

```
cd /var/tmp
unzip 7_Recommended.zip| tar xfp -
cd 7_Recommended
./install_cluster -nosave
```

after rebooting, remove install file with : rm -rf /var/tmp/7_Recommended*

Fix errors in passwd files (Refer to output of Tiger scan in Appendix B)

- Using the output from the Tiger system integrity check, identify accounts with UID=0. For non-root accounts, either change UID or delete account. The account with UID=0 is smtp: since this account is not needed, it can be deleted from */etc/passwd* file.

- Using the output from the Tiger system integrity check, identify accounts with no password and delete from */etc/passwd* file.

- Using the output from the Tiger system integrity check, identify accounts disabled accounts with valid shells. In */etc/passwd* file, change the valid shell to /dev/null.

setuid/setgid programs and files with known vulnerabilities

- Using the output from the Tiger system integrity check, identify files with incorrect setuid/setgid attributes and take appropriate step to correct indicated vulnerability.

Change ownership of system files to root

- Using the output from the Tiger system integrity check, identify files or directories that should be owned by root but are not. Change the ownership of the files using the command:

```
chown root /filename  -or- chown -R root /directory
```

File systems are group writeable

- Using the output from the Tiger system integrity check, identify files or directories that are listed as being group writeable. Use the chmod command to set file access privileges appropriately.

Enable logging to allow minimal intrusion detection

- Create and enable a log file to record all system messages of priority auth.info or higher. Edit the */etc/syslog*.conf file to include the following line (where whitespace are Tabs):

```
auth.info	/var/log/authlog		# to logfile
```

Create the log file and set proper permissions on the file with these commands:

```
touch /var/log/authlog
chown root /var/log/authlog
chmod 600 /var/log/authlog
```

Stop and restart the syslogd so that logging is enabled:

```
/etc/init.d/syslog stop
/etc/init.d/syslog start
```

It is also recommended that some shell script be installed to rotate logs on a regular basis so that individual log files do not grow too large. Additional logging will also be required but these procedures will not be covered here.

Disable generation of core files where not required

- Inventory server - No development work is done on this server. Disable the generation of core files entirely by adding the following line to */etc/system*:

```
set sys:coredumpsize = 0
```

- Parts server - Software development on this system is ongoing and core files are used as a debugging tool. Eliminate core files on a per user basis by adding the following line to */etc/.login*:

```
limit coredumpsize 0      # csh
```

Allow developers needing core files to re-enable core file generation in the .cshrc file in their personal account.

- To ensure that core files do not remain on the Parts server for extended periods of time, execute the following command from root's cron file to delete existing core files that have not been accessed in the last two days:

```
0 2 * * 0 find / -name core -atime +2 -exec rm -f {} ';'
```

Prevent user programs from causing deliberate buffer overflows

- User programs may cause a buffer overflow by attempting to execute code out of the stack. This action would result in a core file being generated which an attacker could then use. The preceding directive disables core files but it is good practice to also minimize the possibility of buffer overflows. To provide additional protection, add the following line to */etc/system*:

```
set noexec_user_stack=1
```

Prevent unauthorized access to system boot process

- To avoid a potential for a denial of service attack using the system boot process, EEPROM security mode should be set to the value 'command'. Additionally, setting this value forces a password to be entered when an attempt is made to run any EEPROM commands other than the normal boot process. Setting this value requires the operator to choose a password. Use caution when saving the password because it MUST be used to allow future access to the system. The command needed is:

```
eeprom security-mode=command
```

Additional recommendations to minimize exposure to known vulnerabilities

- The scan by Tiger also reports other problems with the filesystem such as files being unowned or directories that are world writeable. After all the more serious problems have been addressed, these items should be resolved as appropriate.

- System run states other than level 2 are not used. The links that start services at these run levels can be safely removed with the following command:

```
rm -f /etc/rc[ 013] .d/*
```

- Until an application like SSH is implemented, minimize the potential impact of .rhosts files. Run this command from root's cron file nightly to remove .rhosts files created by users:

```
15 2 * * 0 find /name .rhosts -exec rm -f {} ';'
```

- X-Windows is not required on the Inventory server. Remove the xhost command from the server (rm /usr/openwin/bin/xhost) and add a line to root's cron file to search for and delete any xhost commands installed in user accounts:

```
16 2 * * 0 find /name xhost -exec rm -f {} ':'
```

- Since FTP is required, its should be restricted to known, valid users. A */etc/ftpusers* file should be created that contains all of the names of accounts NOT allowed to use FTP. To create the file, use these commands:

```
touch /etc/ftpusers
for user in root daemon bin sys nobody noaccess nobody4 uucp \
        adm lp smtp listen
do
        echo $user >>/etc/ftpusers
done
chown root /etc/ftpusers
chgrp root /etc/ftpusers
chmod 600 /etc/ftpusers
```

- Develop local policy and methods to download and install Solaris security patches as they become available.

- A scan of the local area network using the nmap network scanning tool reveals other servers residing on the local net (See Appendix A). There may be configuration vulnerabilities in these systems which would allow them to be used as a base of attack for the OIS or other servers. After both OIS servers have been reasonably secured, it is worthwhile to conduct security auditing on all other identified system on the local net.

B. Solutions to documented vulnerabilities requiring additional costs, significant system reconfiguration and/or system downtime:

Before addressing individual system vulnerabilities, systems administrators must face the possibility that the servers have already been compromised by attackers. Minimal security was installed initially and no logging was ever done to detect intrusions. Implementing all of these recommended changes for a compromised system will still result in an unsecured system.

At the earliest possible opportunity, system administrators need to take both OIS servers off-line, backup critical data and applications, and reinstall the operating system from original distribution media. Starting from this 'clean' install is the only way to guarantee that systems have not been compromised before implementing additional security features. Furthermore, it is recommended that the administrators obtain a copy of the Solaris Security Step by Step Version 1 guide from the SANS Institute (www.sans.org) and follow the procedures outlined for installing a secure Solaris operating system.

All discussions in section B will assume that the operating systems were re-installed and the operating systems on both servers are clean and secure. Since most of these recommendations involve the installation of new software applications, the individual "How-To's" and configuration instructions will not be discussed here. Instead, please refer to the documentation included with each application for installation and configuration instructions.

- <u>Several user authentication and password issues must be addressed</u>:

1) username/passwords transmitted over the network in the clear,
2) easily guessed passwords and
3) restricting user access to known IP addresses.

In the current implementation of the OIS, there are some major vulnerabilities with user accounts and passwords. Unfortunately, due to the nature of the client/server application being used, not all of the identified problems can be addressed. (Please see Section C immediately following where this problem is discussed.) But there are three things that can be done now to address these issues.

The easiest of the three issues to deal with is the problem of easily guessed passwords. Installing the passwd+ version beta 0.7 application by Matt Bishop (ftp://nob.cs.ucdavis.edu/pub/sec-tools/passwd+.beta.tar) will force users to select better passwords for their accounts. Passwd+ proactively checks new passwords against dictionaries and any number of locally chosen tests and rejects a password if it is too easy to guess. Used along with standard password aging, this application will greatly minimize the use/reuse of poor passwords.

The two remaining issues can be partially addressed by the installation of Secure Shell (SSH) and TCP Wrappers. SSH version 1.1.27 (http://www.ssh.fi/sshprotocols2/download.html) is an application designed to replace the rsh/rlogin/rcp functions native to Unix. SSH uses different types of authentication to create an encrypted 'tunnel' for communications between networked systems. The SSH application can also be configured to support FTP and X-Windows sessions. TCP Wrappers (ftp://ftp.porcupine.org/pub/security/index.html) is an application which allows network connections to a host system based on the IP address of the remote system. Additionally, TCP Wrappers can be configured to log all rejected connections to provide increased monitoring capabilities.

TCP Wrappers can easily be installed and configured on the OIS. Since all users of the OIS are known, a */etc/hosts.allow* config file can be constructed with entries for all known remote system. For example, to allow connects from a system on the local network, the line

```
ALL : 123.123.20.41
```

in */etc/hosts.allow* would permit all connections to the OIS from that machine. After the .allow file was configured, an */etc/hosts.deny* file would be created with a single line:

```
                    ALL : ALL
```

The result is that any remote system not specifically referenced in the allow file will be denied access to the OIS.

Installation and use of SSH is more of a problem. First, since current client/server applications will not work with SSH, the majority of communications to the OIS will bypass SSH. Second, there are licensing fees involved with the commercial use of SSH which may not prove cost effective for a lightly used program. Finally, SSH requires each remote system to be running a SSH compatible communications program. While use of the remote program can be required for in-house use, OIS users at other organizations may not be willing to install the necessary software. As a result, SSH may not provide enough overall security to be worth the effort needed to install and support it.

- Web server application and locally developed CGI applications are not secure.

The information available from the OIS web server is the most sensitive data in the entire OIS system. This information has to be protected in order for the company to maintain its competitive advantage. One of the major efforts that must completed before the OIS is made accessible to the Internet is to completely overhaul the current web implementation.

The web server now running is an old NSCA application that is quite insecure. At a minimum, the latest version of the Apache web server (version 1.3.12 http://www.apache.org/httpd.html) should be installed and secured. Several security features can be implemented. First in terms of general systems security, Apache can be installed in a chroot()ed environment. This provides some additional protection to the Parts server in the event the web server application is compromised by an attacker. If implemented correctly, only the directories containing the web server will be vulnerable to the attack, not the operating system or other applications on the Parts server. Second, configuration files permit allow/deny type setup where access can be granted or restricted by individual hosts. Configuration parameters can also permit or deny access to different server files and directories. It is also possible to require users enter passwords to access certain information. Taken as a whole, the configuration files allow a high degree of granularity when setting access to data. Finally, since data should be protected while in transit over the network, Apache is capable of using Secure Socket Layer (SSL, available from http://www.openssl.org) to encrypt data. This will help minimize the threat from network sniffers.

In additional to installing a new web server, all of the locally written code in the custom CGI programs should be reviewed and fixed as necessary. Weak CGI code is the source of many web server compromises, mostly from failure to check user inputs to CGI scripts. Existing code should be checked for common errors such as failing to check input value string lengths or string content. Given the importance of protecting web accessible data, if there are no CGI 'expert' programmers on staff able to adequately check existing code, it might be worthwhile to hire or contract out for a person who can accomplish the task.

- System subject to common RPC attacks.

The need to run RPC commands in the OIS opens the systems to well known RPC attacks. However there is one big advantage in that the functionally of the OIS is not dependant on either NFS or NIS/NIS+. Ensuring these services are disabled blocks a number of common attacks. One other factor in favor of the OIS is that all allowed RPC requests should originate either from the Inventory server or from the Parts server, never from remote systems.

This situation should allow the OIS to benefit from the installation of the rpcbind replacement application by Weitse Venema (ftp://ftp.porcupine.org/pub/security/index.html). This version of rpcbind allows access to be filtered by IP address. By configuring each server in the OIS to only accept RPC requests only from its partner server, potential attacks from other machines can be minimized. (Also, please see the discussion below regarding hardware issues. Combining the functions of the Inventory and Parts servers on a single platform would eliminate the need for RPC entirely.)

- No user disk quotas resulting in potential denial of service if volumes fill to capacity.

The current implementation of the OIS allows users to generate large reports in their home directory on the server. There is the possibility, either unintentional or intentional, of a user creating enough sizable reports to fill the user disk partition. To prevent a denial of service due to a full partition, quotas should be enabled on the /export partition containing the user accounts. To enable quotas, first create a quota summary file using the commands:

```
cp /dev/null /export/quotas
chmod 600 /home/quotas
chown root /home/quotas
```

Then mark the partition as having quotas enabled by modifying the */etc/vfstab* file. This is done by adding the option 'rq' to the /export partition. Quotas are then enabled by the command:

```
quotacheck -a
```

Finally, set quotas for all user accounts using the command:

```
edquota username
```

- No hardware redundancy, multiple points of failure could cause system downtime.

As noted, the system hardware used to support the OIS has multiple components that might fail and cause the servers to go off-line. Also as noted, one of the servers is almost four years old and reaching the end of its useful life. In the near future, the server will probably need to be replaced. In the mean time, OIS system administrators should consider instituting accounting procedures on both servers. Not only will accounting functions provide additional logging information, they will provide important usage data which can be used to help choose a new server.

If the budget permits and system usage warrants, strong consideration should be given to replacing the current servers with a single server having enough capacity to support both the Inventory and Parts function. A single server would provide the following benefits:

- Newer systems can be purchased with redundant components and hot-swappable drives to minimize hardware failures causing system downtime.
- Both server functions on the same system will eliminate the need for using network RPC calls for process communication eliminating a major security issue.
- Consolidation of functions on a single system would free one of the existing severs to be used as a development platform. This would remove the requirements to maintain developer tools and accounts on an Internet accessible server, closing another big security hole. It would also mean that new applications or CGI programs could be thoroughly tested in-house prior to making them available to end-users.

C. Suggestions to minimize the identified vulnerabilities which are currently unresolvable or require modifications to vendor applications. Items of low priority are briefly discussed as well:

- Passwords in client/server application vulnerable to network sniffing attack.

The single most important vulnerability that cannot be immediately remedied is the fact that the client/server application used for system access passes account names and passwords across the network unencrypted in the clear. Any attacker with a network sniffer could intercept passwords and use them to attack the OIS. In preliminary communications with the vendor via email, it was determined that current encryption methods are not supported in the client and that the client implementation used will not function using secure shell (SSH) technology. However, the vendor did indicate a willingness develop custom code if a workable solution can be found. Additional technical discussions and negotiations will be needed to develop a reasonable solution.

As a starting point for technical discussions with the vendor, a process simulating one-time passwords should be proposed. If it is possible for the vendor to include an encryption algorithm as part of both an updated client application and server application, system logins could be managed using encrypted strings rather than passwords. There are already several OIS system characteristics in place that favor this type of approach:

- All users must access the OIS through a custom client application available only from the OIS administrators. Using one-time password terminology, this is the "something you have" component.
- When the user's account is established, two files are installed on their PCs; the client application and a small initialization file telling the client how to find the server. At the time these files are placed on the client, the system administrator can generate a unique character string/identifier for the user. This string can then be stored in two places, first in the client's initialization file and second on the server in a protected 'lookup' file. In the server lookup file the string

would be linked to the newly established username and given protections similar to a password file. Again referring to one-time password terminology, this string would be the "something you know" component.

- The third characteristic in favor of this solution is that all user interaction to the OIS is through the client application. For normal operation, users need never to open a telnet session to access data. If minor modifications are made to the applications, all output can be directed to the user's PC eliminating the need for ftp access to the server.

Using these features, client login requests can then proceed along these lines (client actions in italics):

- *Client requests login screen from server*
- Server responds with a request for client's username
- *Client responds to request with username*
- Server uses client's username to search the 'lookup' file to find the previously established unique character string. If the username is not found, the connect request fails immediately. If the username is found, the server generates some random token/string, possibly based on the current system time. This token is sent to the client.
- *The client receives the token and concatenates it to the unique character string stored in the local initialization file. The resulting string is encrypted using the algorithm built into the client and the encrypted string is sent back to the server.*
- The server temporarily stores the encrypted string. Since the server knows the stored unique character string and the random token, it can do the same encryption as the client. If the resulting encrypted strings from client and server match, the user is authenticated and the normal session begins. If they don't match, the connect request fails.

As always, there are both advantages and disadvantages to this scenario if it is feasible. The two biggest limitations are that 1) only the connect string is encrypted not the entire session, and 2) it is still possible to break the encryption on the passed string. While a fully encrypted session is desirable, it is not absolutely necessary. The information coming from the inventory database is of minimal value to an attacker if it is intercepted. The more critical information is stored in the Parts database and viewed through a web browser. If the parts information can be viewed from a secure web connection using SSL, the overall security of the data is acceptable.

But there is one big advantage if this scenario will work. Since the user only needs access to the server application after authentication, the shell for the users account can be disabled by setting it to /dev/null. So even if an attacker could break the user login, they would never get a login shell to use. This leads to an enormous security benefit in that the user accounts needed to access the servers using telnet can be limited to a hand full of administrative and developer accounts. Theses few users do not use the client application so they can easily be configured to use secure shell accounts.

- OIS administrators have no control over corporate firewall configuration.

Unfortunately, there could be many issues that might prevent effective communication and cooperation between OIS administrators and the corporate IS group managing the firewall. One of these may simply be that IS has a view of the world that is Microsoft Windows-centric and just failed to consider the issues involved in having Unix systems on the internal network. At a minimum, OIS administrators should prepare a document/memo to IS containing two prominent sections. The first section should summarize the damage that could be done by a compromised Unix server on the network as well as the efforts under way to make the existing systems more secure. The second section should provide a detailed list of exactly what needs to be done to modify the firewall. Some of these action items might include:

- prevent packets from outsiders that appear to be from the local network
- block common ports used by X-Windows
- block access to port 111 to prevent external access to portmapper
- block access to ports in 32K range to prevent access to RPC services running on Solaris
- block port 2049 and 4045 to prevent access to NFS and lockd, respectively.

In generating this memo, IS should be reminded of the consequences of doing nothing but should also be given a head start to fixing potential weaknesses. At the very least, documentation in the form of this memo to company managers should raise the issues to a level of awareness where something will be done.

- Low priority vulnerabilities that can be ignored with minimal risk

- Computer room door is insecure

The computer room is located in an inner room of a normally staffed office in a building that is monitored 24 hours a day. The threat from outside intruders is low. If physical security needs to be improved, the only additional item needed would be a dead-bolt lock with a different restricted key set.

- Vendor technical support available only during normal business hours.

The majority of system access occurs during normal business hours, as does routine maintenance. As the system use is expanded, system activity levels should be monitored to determine at what point usage dictates that all maintenance be confined to off-duty hours. At the point when significant systems activity will be occurring on nights and weekends, management can be approached for additional funds for 24 hour technical support.

- Network cabling is not secured from unauthorized access or tampering.

To the present time, the assumption was the firewall and the actions taken by the corporate IS department was adequate to protect the network. 24 hour monitoring of the building again minimizes outside threats. Also, it was assumed that authorized corporate users of the OIS should have complete access to all data so internal physical threats are minimal. In terms of costs versus benefits, the large amount of money needed to secure existing cabling would return few benefits.

- Network communications equipment is not protected by backup power supplies.

This is of minor concern since the OIS is not at this point considered a 'critical' system and generates no direct revenue for the company. If communications are lost due to a power failure, users outside of the company are not greatly inconvenienced if the OIS is unreachable for short periods of time. The more important concern has already been addressed, sufficient backup power to enable the server to be taken off-line gracefully without damage or data loss.

References & Software Sources:

Editors, Sys Admin Magazine. Sys Admin Essential Reference Series: UNIX Security. R&D Books, 1997, ISBN:0-87930-471-5.
Frisch, AEleen. Essential System Administration, 2nd Edition. O'Reilly & Associates, 1995, ISBN:1-56592-127-5.
Garfinkel, Simson, and Gene Spafford. Practical UNIX & Internet Security, 2nd Edition. O'Reilly & Associates, 1996, ISBN:1-56592-148-8.
Nemeth, Evi, Garth Snyder, Scott Seebass, and Trent R. Hein. UNIX System Administration Handbook, Second Edition. Prentice Hall, 1995, ISBN:0-13-151051-7.
Pomeranz, Hal, Ed. Solaris Security Step by Step, Version 1.0. The SANS Institute, 1999.
Pomeranz, et al. SANS DC2000 Security, UNIX Security Course Notes. The SANS Institute, 2000.

Apache web server : http://www.apache.org/httpd.html
crack version 5.0 : ftp://coast/cs/purdue.edu/pub/tools/unix/crack/crack5.0.tar.gz
nmap version 2.53 : http://www.insecure.org/nmap/
passwd+ version beta 0.7 : ftp://nob.cs.ucdavis.edu/pub/sec-tools/passwd+.beta.tar
rpcbind : ftp://ftp.porcupine.org/pub/security/index.html
Secure Shell (SSH) version 1.1.27 : http://www.ssh.fi/sshprotocols2/download.html
Secure Socket Layer (SSL) : http://www.openssl.org
TCP Wrappers : ftp://ftp.porcupine.org/pub/security/index.html
tiger version 2.2.4 : ftp://net.tamu.edu/ftp/security/TAMU/tiger-2.2.4p1.tar.gz

Appendix A - Sample Output from nmap Network Scanning Tool

```
# ./nmap -sT -sR 123.123.20.50
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on abcparts (123.123.20.50):
(The 1496 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)
7/tcp      open        echo
9/tcp      open        discard
13/tcp     open        daytime
19/tcp     open        chargen
21/tcp     open        ftp
23/tcp     open        telnet
25/tcp     open        smtp
37/tcp     open        time
79/tcp     open        finger
111/tcp    open        sunrpc (rpcbind V2-4)
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
515/tcp    open        printer
540/tcp    open        uucp
1103/tcp   open        xaudio
4045/tcp   open        lockd (nlockmgr V1-4)
6000/tcp   open        X11
6112/tcp   open        dtspc
7100/tcp   open        font-service
32771/tcp  open        sometimes-rpc5 (status V1)
32772/tcp  open        sometimes-rpc7 (rusersd V2-3)
32773/tcp  open        sometimes-rpc9 (kcms_server V1)
32774/tcp  open        sometimes-rpc11 (cachefsd V1)
32775/tcp  open        sometimes-rpc13 (ttdbserverd V1)
32778/tcp  open        sometimes-rpc19 (dmispd V1)
32779/tcp  open        sometimes-rpc21
Nmap run completed -- 1 IP address (1 host up) scanned in 63 seconds
```

```
# ./nmap -sP 123.123.20.12-49
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Host abcmodel (123.123.20.17) appears to be up.
Host 500_2_HP4000N (123.123.20.20) appears to be up.
Host  (123.123.20.21) appears to be up.
Host  (123.123.20.24) appears to be up.
Host  (123.123.20.30) appears to be up.
Host abcoctane (123.123.20.31) appears to be up.
Host  (123.123.20.34) appears to be up.
Host  (123.123.20.38) appears to be up.
Host abcterp (123.123.20.39) appears to be up.
Host abcnonmem (123.123.20.40) appears to be up.
Host  (123.123.20.41) appears to be up.
Host abcimage (123.123.20.48) appears to be up.
Host abcinvent (123.123.20.49) appears to be up.
Nmap run completed -- 49 IP addresses (13 hosts up) scanned in 26 seconds
```

Appendix B - Examples of Output from Tiger System Integrity Scanner

(Output was severely edited to save space. What is shown here is representative of the type of problems found.)

```
Security scripts *** 2.2.3, 1994.0309.2038 ***
Mon Aug  7 10:29:19 EDT 2000
10:29> Beginning security report for abcparts (sun4u SunOS 5.6).
# Performing check of passwd files...
--WARN-- [pass002w] UID 0 exists multiple times in /etc/passwd.
# Performing check of group files...
# Performing check of user accounts...
# Checking accounts from /etc/passwd.
--WARN-- [acc011w] Login ID user1 does not have a password.
--WARN-- [acc012w] Login ID smtp has uid == 0.
--WARN-- [acc001w] Login ID adm is disabled, but still has a valid
shell
         (/bin/sh).
# Performing check of /etc/hosts.equiv and .rhosts files...
# Checking accounts from /etc/passwd...
# Performing check of .netrc files...
# Checking accounts from /etc/passwd...
# Performing check of PATH components...
# Only checking user 'root'
--WARN-- [path002w] /usr/sbin/accept in root's PATH from default is not
owned
         by root (owned by lp).
# Performing check of anonymous FTP...
# Performing checks of mail aliases...
# Checking aliases from /etc/mail/aliases.
# Performing check of `cron' entries...
# Performing NFS exports check...
--WARN-- [nfs007w] Directory /usr/man exported R/O to everyone.
# Performing check of system file permissions...
--WARN-- [perm019w] /etc should not have group write.
--WARN-- [perm003w] /export should not have group write.
--WARN-- [perm001w] The owner of /usr/ucblib should be root (owned by
bin).
# Performing signature check of system binaries...
--ERROR-- [init005e] Don't have required file SIGNATURE_FILE.
# Checking for known intrusion signs...
# Performing check of files in system mail spool...
# Performing system specific checks...
# Performing checks for SunOS/5...
--WARN-- [no-id] The PROM monitor is not in secure mode.
--WARN-- [misc008w] NFS port checking disabled in kernel.
```

```
# Running './scripts/check_sendmail'...
# Checking sendmail...
# Checking setuid executables...
--FAIL-- [fsys001f] File /etc/lp/alerts/printer is a setuid script:
-r-sr-xr-x   1 lp       lp           203 Jul  2  1997
/etc/lp/alerts/printer
--WARN-- [fsys002w] setuid program /opt/SUNWvts/bin/ptexec has relative
         pathnames.
--WARN-- [suidxxx] Setuid file `/usr/openwin/bin/sys-suspend' which is
group
        `bin' writable.
# Checking setgid executables...
--CONFIG-- [fsys003c] No setgid list... listing all setgid files
--ERROR-- [init005e] Don't have required file SIGNATURE_FILE.
# Checking unusual file names...
# Looking for unusual device files...
# Checking symbolic links...
# Checking for writable directories...
--INFO-- [fsys008i] The following directories are world writable:
/export/home/chocks/
--WARN-- [xxxxx] The following files are unowned:
/export/home/connollm
# Performing check of embedded pathnames...
```