Global Information Assurance Certification Paper

SANS UNIX TRAC Security Practical

Sunday, January 16, 2005

Matthew R. Versaggi

Sanitized Version

HPI-INC.NET

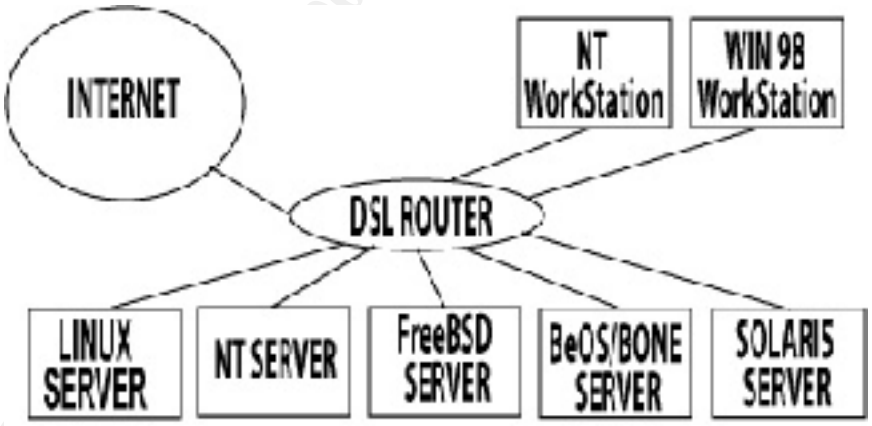
Security of a Solaris Machine on an Experimental Network

A UNIX and Network Security Audit
Analysis, Results, and Recommendations.

EXECUTIVE SUMMARY

Introduction

This audit report focuses on the security aspects of a Solaris machine on the HPI-INC.NET experimental network. Pertinent security aspects of the overall network will also be discussed when appropriate. The HPI-INC.NET network's primary purpose is to provide a stable, secure, research and development platform. It's a small network, consisting of 5 experimental servers and 2 development hosts, all of which are connected to the Internet via a dedicated DSL connection as shown below.



There are only a few accounts (less than 10) which can be accessed from any host console(s) or remotely. The general security philosophy is to try to keep intruders out of the HPI-INC.NET hosts, prevent denial of service attacks, and protect the integrity of communication within the network. This audit interrogates all significant facets of the HPI-INC.NET Solaris server, which would affect its security level including security policy and documentation, physical security, security architecture and design, host security, network security, third party application vulnerabilities, and the SANS top 10 vulnerabilities checklist (excluding Windows).

The Solaris machine is a SUN Sparc 10 running SunOS 5.7 that has existed live on the network for 6 months since the "out of the box" installation. The other servers existing on the network have been "hardened" as specified by the appropriate SANS Step-by-step guides (Linux and NT), FreeBSD Security How-to, and the BeOS Security Article (see references) within the last 3 months. The Solaris server has *NOT* been hardened as per the recommendations of the SANS Solaris Step-by-step guide, however some non-documented steps were taken to insure the integrity of the machine.

Major Problems
There are many, many glaring security problems existing on this Solaris server. The basis of those problems are a utter lack of any written security policy which would have driven attention and action into fortifying the mechanics of the HPI-INC.NET network site and this Solaris server. There is no disaster recovery mechanisms in place, a badly flawed logical security architecture and design, very limited network based security to speak of while on the backdrop of loads of network based security vulnerabilities. There is almost no 'effective' host based security, while again, on the backdrop of many clearly identified host security vulnerabilities. To top it off, every single one of the SANS top 10 UNIX based security vulnerabilities exists on this Solaris machine in some form. There are no firewalls, no network address translation being done, and no host or network based vulnerability testing going on whatsoever.

Conclusions and Recommendations
The good news is that there concrete evidence to indicate that no one has broken into this system just yet! WooHoo! ;-)

It is highly recommended that HPI-INC.NET implement a sound security policy, construct a reliable disaster recovery system, erect a robust firewall, correct their logical security architectural flaws, manifest solid host and network based security mechanisms, and institute thorough host and network based vulnerability testing immediately. The details of just how to accomplish these tasks are articulated in the remainder of this document.

SECURITY POLICY AND DOCUMENTATION

Introduction
It's clearly understood that the written documents and security policies of an organization are "the" most important components of an organizations overall security policy. It serves to educate and empowers users to maintain a safe stable computing environment. It safeguards users when disaster strikes and provides for the who, what, when, where, and how of responsibility for taking action when an intruder gets in. These concepts, however, haven't penetrated the VIS, INC., establishment just yet.

Summary
HPI-INC.NET has no written security policy or documentation of any kind, thus the formal dissemination of responsibility / accountability and empowerment, which naturally accrues from a formal security policy documentation is non-existent and therefore non-effective. Some positive observations are that there are "implied" and "understood" security procedures, which seem well followed. Ideally, these would be codified.

Recommendations
HPI-INC.NET must construct a formal security policy ASAP, which is complete and thorough covering the items listed in the *'Specific Critique'* section below. This information should serve as

a basis for preventative, disaster recovery, and incident handling situations moving forward. It is highly recommended that the GIAC Basic Security Policy V1.34 is referenced as a model for any of VIS, INC's security policy construction activities.

Specific Critique

- Admin Practices
  There are no written documents whatsoever articulating any security policy, responsibility, accountability, disaster recovery, user monitoring, or legal consequences from misuse of the system.

- Who can use what resources
  Since there are no written documents, there are no specifications as to whom is authorized to use what resource. Users are presumed to have enough intelligence and moral fiber to not touch "things" which do not belong to them or that could harm the system.

- Proper Use
  There is no written procedure explaining the proper use of hardware or software on this network. If a user runs a piggish program or bandwidth hog, which bogs down the system or network, the current absence of a usage policy permits this by default.

- Granting Access and Use
  This item, although not written down is very clearly articulated and understood in the organization: No one can have access to the experimental network in ANY way unless they get explicit authorization from the senior manager / sysadmin.

- User Rights and Responsibilities
  This are is obviously lacking (since it doesn't exist) and should articulate what users should and shouldn't do, define what unacceptable behavior is, what to do when user passwords are forgotten, etc.

- Sensitive / Proprietary Information
  Again, there is no policy regarding sensitive information in the network. It is presumed that users are morally righteous and upstanding enough to not pry into others information. The experimental servers in general have no sensitive information on them, however the development boxes do.

- System Security Configurations
  Again, there is nothing written down at all. Ideally, information should exist articulating such things as using TCP Wrappers, and SNORT as a mandatory practice. It should also articulate turning off unnecessary or vulnerable services, invoking proper log configurations and rotations, firewall information and responsibilities, etc.

- Anti-Virus Policy
  Again, nothing written, but in this case a lot is understood and practiced. Anti-Virus and Personal Firewall protection is mandated on every machine and the firewall rules and virus definitions are kept current. Ideally, this could be codified and this responsibility delegated to an individual for its upkeep, and incident handling procedures.

- Password Policy
  Like the Anti--Virus policy, using strong passwords are common, but the codification of that information is not. Procedures for the periodic auditing of password files with CRACK, and 10phtcrack do not exist.

- Backups, Disaster Handling, and Incident Handling

There is an implicit presumption that the sysadmin will be responsible for all of these items but (surprise, surprise) he isn't performing those functions. Bigger surprise, there is no written reference to whom is responsible for what in these instances.

PHYSICAL SECURITY

Introduction
Security considerations often overlook the critical fact that when physical access is attained to the console of any standard UNIX machine, gaining root privileges is a foregone conclusion. To protect the console is paramount and must be done at a number of levels. In addition to this are the considerations of disaster recovery, UPS fortification, locked box access, and environmental controls.

Summary
While not pristine, the physical security of the HPI-INC.NET Solaris machine works well enough. There does exist UPS fortification and power filtering to help cope with power spikes, brownouts, and blackouts. The entire experimental network (Solaris machine included) is physically located in a secure area which very few people are permitted access to at all and of those, only individuals who have high security clearance. The environmental controls are adequate, however there are no fire controls, other than fire detection mechanisms. One of the big problems is the absence of physical backup devices.

Recommendations
The lack of any physical devices for obtaining backups is the most critical issue looming in this area of the audit. VIS, INC., must correct this problem now if they hope to have piece of mind moving forward. The other looming concern is the existence of the machine on a live network for 6 months without hardening. This must be corrected right away. It should be hardened in the manner described in the SANS Solaris Step by step guide. Other lesser issues of the PROM and OpenBoot procedure should be interrogated fully again just to make sure its performing in a way that it is intended to. The security issues surrounding the CDRom should be looked into but considering the security of the venue it's not a high priority issue.

Specific Critique

- PROM and OpenBoot
  Openboot is the firmware that is run on all Sparc PROM's (Programmable Read Only Memory), and must be kept secure since anyone getting access to the console keyboard of a standard Solaris box can issue a "STOP-A" and effectively wreek havoc on the system as well as take complete control of the system. A password scheme can be levied (choices: none, command, and full) for the Openboot process can be set and indeed is set on this Solaris machine (Command level scheme).

- CDRom, Floppy, and Other bootable media
  This Solaris system should be configured to boot from an internal disk rather than external device on a SCSI bus. This Solaris system will boot from an external CDRom drive, which introduces a security risk of the physical access to the machine is ever compromised. Also, it is recommended to add the following lines to the "/etc/rmmount.conf" file to disallow SetUID programs from being brought in on removable media.

```
mount hsfs –o nusuid
mount ufs –o nosuid
```

- Screen Locking
  There is a password protected screen saver with a low timeout in existence. This is a good thing.

- System Recovery
  There are no physical mechanisms for automated backup or recovery. There has been no image taken of the system as it stands in its current state today. This is a serious problem. There is no way to reconstruct the system in the event of catastrophe.

- System Build Audit
  The system build was done on a SPARC 10 machine *while* having a live DSL connection to the Internet. Appropriate actions were taken in terms of selecting passwords, but little else was done to secure the system after the out of the box installation. It remains that way to date.

SECURITY ARCHITECTURE AND DESIGN

Introduction
The security architecture and design of a network site refers to the overall layout of the network, trust relationships, and connections to the outside networks rather than any single particular technical detail or specific hardware / software concern. Even if all of the technical details appear pristine, there still may be logical flaws in the security architecture and design, which may permit vulnerabilities to exist.

Summary
There are indeed alarming security flaws permeating the logical design and architecture of the HPI-INC.NET network site. These must be fixed immediately. The most notable flaws are the Trust relationships between hosts on the network and outside of the network, and the absence of any firewall or packet filtering capabilities between the LAN and the Internet.

Recommendations
A firewall or packet filtering system needs to be erected immediately between the HPI-INC.NET LAN and the Internet. The rules set need to be thoroughly inspected, and kept up to date. NAT (Network Address Translation) needs to be implemented wherever possible. The trust relationships between the hosts on the LAN should be interrogated, tightened up and redesigned to minimize the between host trust factor. Samba should be limited to hosts behind the firewall as well as all windows file and print sharing hosts.

Specific Critique

- Basic Design
  HPI-INC.NET's security architecture is relatively simple, which is good. There are 7 hosts, of which 5 are servers and 2 are development hosts. The protection scheme, dictated by the behavior of the individual hosts within the network, does not include any firewall or packet filter between the network and the outside Internet. All of the host connections are done via straight IP connections through the DSL router. There is no network address translation or IP masquerading being done. This is a BIG problem.

- Trust Relationships
  The bright spots are that all of the UNIX machines (Solaris excluded) have been "hardened" so their Berkley R-Commands have been disabled, and strong passwords are required everywhere. *However*, there is a great deal of trust between machines on this network. Each machine knows about each other via their HOSTS files and the TCPWrappers installed on each of UNIX servers and the BeOS/BONE Server all recognize and trust each other, the development boxes, and an "***outside***" server.

  There is an instance of Samba running between the NT Workstation host and the FreeBSD server. Each UNIX server has an instance of X-Windows running (including the Solaris server). There are file and print sharing capabilities between the Windows development hosts and the Windows NT Server.

  This is definitely not good from a security standpoint, if one hosts gets compromised, it is a foregone conclusion that they all will be compromised. The "/etc/hosts" and "/etc/resolv.conf", "inetd.conf", "hosts.allow", and "hosts.deny" files should be thoroughly interrogated when reworking this trust relationship.

- External Connectivity
  There are no extraneous connections to the VIS, INC network via modems. All connectivity from the outside Internet comes from the dedicated DSL connection, which can be easily monitored and controlled.

NETWORK SECURITY

Introduction
Network security refers to *ALL* of the network services that are run by hosts on the network (such as TELNET, FTP, Sendmail, DNS, etc), any IP filtering or access control (done at either the application, operating system or network levels), and the trust levels between hosts on the network. This is a key area, which has surpassed host based security in importance as the shear numbers of computers being connected together continues to rapidly grow.

Summary
There are some very critical key vulnerabilities existing in this area of the security audit. The conditions described below cannot persist into the future and need to be fixed ASAP. There are issues with trust implementations, lack of firewalls and packet filtering, network configuration concerns, unnecessary and vulnerable services and their corresponding access points (ports), name service, SSH, NFS, SNMP, DDOS, and X-Windows concerns. These are some of the highest priority items on the fix list and deserve immediate attention.

Recommendations
There are many, many items to implement in order to achieve adequate network security to be listed here, so the reader is referred to the below discussion. In addition to implementing all of the recommendations in each of the below sections, it is advised to add the "-t" option to the inetd startup to invoke a trace of all TCP services to the syslogs daemon facility at severity level "notice". Edit /etc/init.d/inetsvc and add this command as the last line:

"/usr/sbin/inetd -s -t &"

Specific Critique

- IP Forwarding, Redirects, Directed Broadcasts, and Source Routing
  To aid in defending against DDOS attacks, there are a number of general security tasks, which must be accomplished; IP forwarding should be turned off and redirects in general should be ignored (receiving and sending) even if the machine is being used as a router. There should be no source routing going on, nor should the system be directing any broadcasts. The system should have a limit on the number of half-open TCP connections on the system, and how long information can live in its ARP cache. There is NO evidence in the "/etc/inet.d/inetinit" file that the system is currently enforcing this the above rules and therefore the following lines should be entered at the end of the file to enforce this policy.

  "ndd –set /dev/ip ip_ignore_redirects 1"
  "ndd –set /dev/ip ip_send_redirects 0"
  "ndd –set /dev/ip ip_forward_directed_broadcasts 0"
  "ndd –set /dev/ip ip_forward_src_routed 0"
  "ndd –set /dev/tcp tcp_conn_req_max_q0 1024"
  "ndd –set /dev/ip ip_ire_flush_interval 6000"
  "ndd –set /dev/arp arp_cleanup_interval 6000"
  "ndd –set /dev/ip ip_forwarding 0"
  "ndd –set /dev/ip ip_strict_dst_multihoming 1"

  There is, however, evidence in the "/etc/inet.d/inetinit" that IP forwarding is turned off during the boot process. However this should be tested for verification purposes. Additionally, the following file should be created.

  "/etc/notrouter"

- Network Configuration Vulnerabilities
  The judicious interrogation of "/etc/inetd.conf" is essential for discovering unnecessary or vulnerable services running on an 'out of the box' installation. This Solaris server was modified to have only the two following lines enabled:

```
ftp     stream  tcp    nowait  root    /usr/etc/tcpd        in.ftpd -l
telnet  stream  tcp    nowait  root    /usr/etc/tcpd        in.telnetd -h –U
```

  Interrogation of these two lines shows some VERY good things, from a security point of view. It illustrates that *TCPWrappers* are indeed being deployed on this machine and are protecting the FTP and TELNET daemons. Interrogation of the daemon flags shows that there is logging of activities of these two daemons occurring, the telnet daemon is being forced to suppress giving out any extra sensitive information about the system to anyone attempting to initiate a telnet connection to the system. The fact that all other services have been turned off is also a VERY good thing from the standpoint of network security.

  Another item on the to do list is to disable the listener on the serial ports by removing the following line from the "/etc/inittab"

  sc:234:respawn:/usr/lib/saf/sac -t 300

- Network Access Control

TCPWrappers: It is a VERY good thing to discover the existence of TCPWrappers guarding network access. However the interrogation of the following important files (hosts.allow, hosts.deny) reveals some concerning facts:

```
HOSTS.ALLOW:
#
# Figure out what local boxes to permit total access from
#
#
ALL:   666.663.659.66:   severity auth.info: allow
ALL:   666.663.659.67:   severity auth.info: allow
ALL:   666.663.659.68:   severity auth.info: allow
ALL:   666.663.659.69:   severity auth.info: allow
ALL:   666.663.659.70:   severity auth.info: allow
ALL:   666.663.659.71:   severity auth.info: allow
ALL:   666.663.659.72:   severity auth.info: allow
ALL:   666.663.659.72:   severity auth.info: allow
#
#
# Permit specific access.
#
in.telnetd:    666.663.659.67, 666.663.659.71, 666.663.659.66: severity auth.info
allow
in.ftpd:       666.663.659.67, 666.663.659.71, 666.663.659.66: severity auth.info
allow
#
#
# catch all state.
#
ALL:   ALL:   severity auth.info deny
```

The H*OSTS.ALLOW* file creates trust among <u>all</u> of the hosts in the local network. This is a BIG problem and should be fixed. No trust what so ever is a much better security policy.

```
HOSTS.DENY:
#
# Permit the local UNIX boxes access, no Microsoft!
#
ALL: ALL: EXCEPT 666.663.659.67, 666.663.659.66, 666.663.659.71,
209.666.225.100
```

The H*OSTS.DENY* file creates trust among some of the hosts in the local network and a host <u>outside</u> of the local network (**209.666.225.100**). Trusting an outside server is a BIG problem and should be fixed ASAP. No trust what so ever is a much better security policy.

FTP Services: The existence and interrogation of the file "/etc/ftpusers" was a encouraging as it revealed a hearty list of those who should never be permitted to FTP into the system: *(bin, daemon, lp, adm, sys, nobody, nobody4, listen, root, uucp, nuucp, guest, anonymous).* There is no anonymous server or login for FTP users on this Solaris server.

".Rhosts and /etc/hosts.equiv" files: There were no instances of any of these files found on this system.

SSH: Currently there are NO facilities for secure remote management of this Solaris server. It is highly recommended that secure telnet and ftp services be constructed ASAP.

- Unnecessary Services
  The disabling of unnecessary or vulnerable services is NOT achieved solely by modifying the "/etc/inetd.conf" file alone on a Solaris Architecture. There are services which fall under this category which are not initiated by the "/etc/inetd.conf" file and are a part of the system startup process. They must be "hunted down" individually in the "/etc/init.d:, "/etc/rc(0-6).d" directories and disabled.

  This Solaris server still currently has the NFS server/client, Automounter, NTP and Print Services running, all of which are unnecessary, and some of which are VERY dangerous security holes. Their Sxxx and Kxxx files should be located in there respective "/etc/init.d:, "/etc/rc(0-6).d" directories and via the "mv' command (*to preserve the symbolic links*), renamed to an Xxxx filename convention. This effectively takes them out of the system initialization and boot up process. Additionally, the NFS server and client must have their corresponding entries commented out of the "/etc/dfs/dfstab" and "/etc/vfstab" files.

- Ports to be examined for blocking
  The reduction or elimination of unnecessary or vulnerable network access points is also a critical preemptive security activity. These can be found in the "/etc/services" file and the questionable ones turned off by commenting out their corresponding entries. Here is a short list of *INBOUND SERVICES* only by *port number / type / name* trio found on this Solaris server which need to be seriously considered for removal (modified per Solaris Security, Peter H. Gregory pg. 161-165)

  - Generally unnecessary or security hazard **but** required for testing....

echo	7/tcp	
echo	7/udp	
discard	9/tcp	sink null
discard	9/udp	sink null
systat	11/tcp	users
netstat	15/tcp	
whois	43/tcp	nicname
hostnames	101/tcp	hostname
exec	512/tcp	

  - Generally unnecessary or security hazard...

daytime	13/tcp	
daytime	13/udp	
chargen	19/tcp	ttytst source
chargen	19/udp	ttytst source
time	37/tcp	timserver
time	37/udp	timserver
name	42/udp	nameserver
bootps	67/udp	
bootpc	68/udp	
sunrpc	111/udp	rpcbind
sunrpc	111/tcp	rpcbind
tftp	69/udp	
rje	77/tcp	
finger	79/tcp	
link	87/tcp	ttylink
supdup	95/tcp	
iso-tsap	102/tcp	

x400	103/tcp	
x400-snd	104/tcp	
csnet-ns	105/tcp	
pop-2	109/tcp	
uucp-path	117/tcp	
nntp	119/tcp	usenet
NeWS	144/tcp	news
login	513/tcp	
shell	514/tcp	cmd
printer	515/tcp	spooler
courier	530/tcp	rpc
uucp	540/tcp	uucpd
biff	512/udp	comsat
who	513/udp	whod
talk	517/udp	
route	520/udp	router routed
new-rwho	550/udp	new-who
rmonitor	560/udp	rmonitord
monitor	561/udp	
pcserver	600/tcp	
nfsd	2049/udp	nfs
nfsd	2049/tcp	nfs
lockd	4045/udp	
lockd	4045/tcp	

- Firewalls and Natd
  There is no existence of any firewall or packet filtering functionality presently on the HPI-INC.NET network site and there should be. Some of the servers will have to implement straight through IP routing, but the others should be doing network address translation at the very least. This should be accompanied by a robust IP filtering scheme sporting a solid and tested rules set. Constructing this is a high priority. The firewall protection will not come from the DSL router (because of the model and firmware vintage) and must come from another machine on the network (possibly the Linux or FreeBSD servers), which could implement ipchains or ipfw.

- Name Services (NAMED)
  This Solaris machine is indeed a name server, and is registered with the NIC as "spiderman.HPI-INC.NET". An interrogation of the "/var/adm/messages" file revealed the following informative line regarding the vintage of the "named daemon" this Solaris box is running:

  "Nov 18 12:01:53 spiderman named[145]: starting BIND 9.0.1"

  It is running the latest BIND 9.X series, which is a very new release of the BIND vintage and is a compete rewrite of the old BIND system. I Consider it a V1.0 and as such should be suspect for programming bugs, security holes, etc.

  Illicit Zone Transfers from DNS: This is a growing issue from sysadmins about concerning permitting "just anyone" to pull a zone transfer from the DNS server, especially private/internal DNS servers as a potential "Mapping Attack". This vintage of BIND permits the directive *"allow-transfer"* which specifies the hosts or networks, which may pull DNS information from this server. This is NOT currently being done and should be incorporated ASAP. I'd also recommend altering the version string that BIND reports when queried to something outlandish. That'll keep em guessin! ;-)

Buffer Overflows: Since this Solaris server is using BIND vintage 9.x it has been indicated that the buffer overflows which have plagued earlier versions are no longer present.

Cache Poisoning: This problematic activity occurs when a name server is tricked into believing erroneous information from querying some evil external source, either effecting a denial of service or an unwelcome redirecting of visitors to unwanted sites. Again, since this Solaris server is using BIND vintage 9.x it has been indicated that the buffer overflows which have plagued earlier versions are no longer present.

- NFS (Network File System)
  This Solaris Server should never run NFS for any reason. However it is indeed running as an our of the box configuration and should be shutdown as prescribed earlier ASAP. It is a MAJOR security hole, which is in desperate need of being plugged.

- Sun SNMP Agent: "mibiisa"
  The mibiisa utility is an RFC 1157-compliant SNMP agent. It should not be run with out proper care and feeding. Currently it is a part of the startup sequence and should be removed in the way prescribed earlier.

- Denial of Service Attacks (DDOS) Vulnerabilities
  This Solaris server has been tested for being an amplification site as per the SANS document, "Testing Broadcast Amplification from Solaris, R1.3" and it is clean. However, NONE of the steps required by the SANS document, "Help Defeat Denial of Service Attacks: Step-by-Step, R 1.41" have been implemented and need to be moving forward. Since there are no firewalls or packet filtering in service currently, it is impossible to implement the SANS DDOS directives. In addition to implementing the recommendations per the discussion of the "IP Forwarding, Redirects, Directed Broadcasts, and Source Routing" section above, fixing this will help defend against DDOS attacks.

- X-Windowing System
  This Solaris server is indeed running CDE and has little reason to. It has well known security vulnerabilities and should be shut down. A command line console is all that is required of a server.

- Network Based Vulnerability Detection and Scanning
  There were no provisions set forth to perform any network based vulnerability checking of the other hosts on the network with such tools as SAINT, NMAP, NESSUS or ISS. It is MOST highly recommended that this type of preemptive testing take place immediately to interrogate the other hosts on the network along with the Solaris server for network vulnerabilities. I did discover a tar file of the latest version of SAINT, so there is some thinking along these lines going on here.

HOST SECURITY

Introduction
Host security examines the problems on the Solaris server related to items like individual user authentication and file permissions, the host filesystem, the startup files and sequence, the system configuration files, and vulnerabilities which might be exploited by intruders who are actually logged into the system. It also examines auditing, logging, recovery, maintenance as well as intrusion detection.

Summary
There are many gaping security holes which became painfully evident during this phase of the audit. There was evidence of a general lack of preemptive security actions which would harden the host appropriately. There are system logging and OS fortifications, which should be erected along with legal banners for various network daemons. The most compelling problems are the lack of host based vulnerability testing and host based intrusion detection processes, in addition to the lack of any backup processes for this Solaris server. There is an absence of secure communications for remote management, and an utter lack of security updates and patches from the OS vendor. This is bad. There were also a host of preemptive audits like user activity audits, open files audits, password audits, which were not done.

Recommendations
The recommendations are simple, all of the below discussions are accompanied by definitive recommended actions. They should ALL be done. The logs and backups are a very high priority, as is are the intrusion detection activities, secure remote communications, and vulnerability testing.

Specific Critique

- Operating System Vulnerabilities
  Partitioning: This Solaris box was not partitioned with security in mind. It did not segment and isolate partitions so that the server will continue to function if a partition gets corrupted or is disabled due to a DDOS attack filling a partition with log messages. This is a historical issue however and cannot be fixed now.

  System File: The following changes should be made to the "/etc/system" file in order to erect an additional security barrier against a variety of attacks such as buffer overruns, core dump attacks, and NFS attacks. Add the following:

```
Attempt to prevent and log stack smashing attacks:
set noexec_user_stack = 1
set noexec_user_stack_log =1

Set various useful parameters to good vaues:
set pt_cnt = 256
set rlim_fd_max =1024
set rlim_fd_cur = 256
set maxprc = 150
set sys:coredumpsize = 0

Force NFS clients to use privileged ports:
set nfssrv:nfs_portmon = 1
setnfs:nfs_portmon = 1
```

  Dealing with .rhosts: RHOSTS files permit root logins from anywhere. To cause the system to ignore ".rhost" style authentication, remove following lines (anything with 'rlogin' in it) from the "/etc/pam.conf" file. Currently this in not being done and it should be. Also, a dummy "/.rhosts" file should be created and set to chmod 400 to keep attackers from dropping in a ".rhosts" file of their own.

```
#rlogin  auth sufficient /usr/lib/security/pam_rhosts_auth.so.1
#rlogin auth required   /usr/lib/security/pam_unix.so.1
#rsh    auth required   /usr/lib/security/pam_rhosts_auth.so.1
```

Cron Jobs: Currently the files "/etc/cron.d/cron.deny" and a "/etc/cron.d/at.deny" do exist and only permit those listed to execute the commands "crontab" and "at" to modify cron jobs. This is a good thing.

- Filesystem, File Permissions, SetUID/SetGID Root files and Hidden Disk Space Audit

Filesystem Security Enhancement: It may be desirable to randomize the filesystem Inode numbers to thwart cleaver hackers who attempt to open and alter files by inode numbers instead of file names using the command "fsirand". NOTE: Any filesystem intended to have "fsirand" run on it should be backed up first.

Default File Permissions and UMASK:
It is recommended that the root umask be changed to 077 or 027 which will ensure that any file that is created by root is not readable or writable by others. The default UMASK for system daemons is not normally set at startup time. This results in files being created by these daemons being world-writable by default. This is a problem and should be fixed. One solution to this is to execute the following script (Credits: Hal Pomeranz)

```
echo 'umask 022' > /etc/init.d/umash.sh
chmod 744 /etc/init.d/umask.sh
for dir in /etc/rc?.d
do
        ln –s ../initd/umask $dir/S00umask.sh
done
```

Disk Quotas: There are no quotas of disk space usage in place so any user can have as much as they want. It is recommended that quotas be established and enforced.

Hidden Space Audit: LSOF found the following files with **link = 0**:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NLINK	NODE	NAME
init	1	root	5u	FIFO	32,24	0t0	0	18904	/etc/initpipe
cron	171	root	3u	FIFO	32,24	0t0	0	34961	/etc/cron.d/FIFO
vold	222	root	6u	VCHR	105,21	0t0	0	32283	/devices/pseudo/tl@0:ticots
dmispd	250	root	5u	VCHR	105,24	0t0	0	32283	/devices/pseudo/tl@0:ticots
dmispd	250	root	6u	VCHR	105,25	0t0	0	32283	/devices/pseudo/tl@0:ticots
dmispd	250	root	7u	VCHR	105,26	0t0	0	32283	/devices/pseudo/tl@0:ticots
snmpXdmid	254	root	2u	VCHR	105,27	0t41	0	32283	/devices/pseudo/tl@0:ticots
snmpXdmid	254	root	3u	VCHR	105,28	0t0	0	32283	/devices/pseudo/tl@0:ticots
snmpXdmid	254	root	4u	VCHR	105,29	0t0	0	32283	/devices/pseudo/tl@0:ticots
sac	260	root	1u	FIFO	32,24	0t24	0	29578	/etc/saf/_sacpipe
sac	260	root	4u	FIFO	32,24	0t8	0	32462	/etc/saf/zsmon/_pmpipe
ttymon	263	root	1w	FIFO	32,24	0t24	0	29578	/etc/saf/_sacpipe

As a general rule all files, which have a link count of zero should be investigated as suspicious.

SetUID / SetGID / World-Writable Files: There is no record of any system check for the existence of user files which have enhanced permissions. There is also no record of an audit checking for world-writable files or directories. Since this is an ongoing security concern, it is recommended that a thorough check be made of the files on the system for the existence of these kinds of files and if any of suspicious nature are found, they should be investigated. The command to search the system for SetUID, SetGID or World-Writable files are as follows:

SetUID Files:
Find / -type f –perm –4000 –print

SetGID Files:
Find / -type f –perm –2000 –print

World-Writable Files:
Find / -type f –perm –o+w –print

Filesystem Auditing Tools: There are no installations of Tripwire, ASET, COPS, or Tiger on this Solaris server, and there is no evidence / record of them ever being run. It is highly recommended that these filesystem auditing tools be procured and run regularly against the file system to help the intrusion detection process and aid in maintaining the integrity of the filesystem.

- System Logs, and Remote Logging
  TCP Trace: It is useful to be redundant in stating that the file "/etc/init.d/inetsvc" should be edited and this command replaced as the last line:

  "/usr/sbin/inetd -s -t &"

  SysLog.Conf: It is recommended that the following commands be added to the "/etc/syslog.conf" and the corresponding files be created via roots 'touch" command in their respective directories. This will ensure a rich set of logs to interrogate moving forward.

auth.*	/var/log/authlog
authpriv.*	/var/log/authlog
daemon.info	/var/log/daemon
mail.info	/var/log/maillog
user.err	/dev/console
user.err	/var/log/messages
*.notice;kern.debug;mail.crit	/var/log/messages

  Remote Logging: A security bonus would be for this Solaris server to perform remote logging and send its logs to a central log host for archiving. It's not a high priority item however.

  System Accounting Information: There currently is NO system process and accounting information being logged and therefore no good benchmarks / baselines of the system which can be used to measure anomalies against. It highly recommended that the "SAR" utility in conjunction with the Solaris products "SUNWaccr" and "SUNWaccu" be used to form a process that is put in place and run on a regular basis so that enough statistics could be takes to form an accurate model of system behavior over time. This is useful for detecting intruders when system utilization behaves in unexpected ways.

  Security Tools: There are no instances of useful system log security tools on this Solaris server and it is highly recommended that at LOGCHECK is incorporated as a logfile analyzing mechanism.

- Backups & Recovery

There are currently NO processes in place for any backup (incremental, differential, or full), no disaster recovery procedure, hardware or software. This is a monster problem which must be resolved immediately.

- Authentication
  Passwords: Although the general consensus is that everyone uses strong passwords, after a judicious application of CRACK on the password / shadow file, it has proven otherwise. It is recommended that a all users be mandated to reset their passwords using a strong password scheme. Additionally, the "/etc/default/passwd" files contains parameters which can turn on "password aging" and other useful things. It is recommended to do this, as it is currently not enforced.

- Sensitive Data
  At the moment, there is no extra encryption mechanisms existing on this Solaris server, such as BlowFish or MD5. Only the encryption processes which exist in an 'out of the box' vintage of Solaris exist. It is recommended to procure a PGP or similar encryption suite for use in safeguarding important information on the file system.

- User Security & Environments
  PATH and LD_LIBRARY_PATH: These environment variables must be kept safe. This means that the directories which they reference should not have 'group' or 'other' write access. Additionally, they should not have directories who's content is questionable. The "." should NEVER appear in the path of any user, or it will open up the possibility for the planting of Trojan Horse programs that wait like land mines for root to trip over. There have been no record of auditing the environment variables of the ".profiles" for the current users to inspect for this anomaly, therefore it recommended to do so soon. TIGER should considered as a audit tool for this task.

- Shells and Secure Shell
  The popular choice for shells is BASH. However, there is exists no infrastructure for a Secure Shell client or server in place at the moment. This is critical for maintaining the integrity of communications during remote management, and is a high priority item.

- Process and Port Audit
  There is no proof of any process or port audit being executed ever, such as the ones achievable with LSOF or NETSTAT. It is highly recommended that LSOF be used to map a baseline of system process behavior, starting with the system open files, open sockets, user open files, and user PID tracking.

- User Activity Audit
  It is highly recommended to do a quick audit of the following user activity log files multiple times on a daily basis. Be sure to adequately secure these log files with the file permissions "-rw-r—r—" because sometimes the default permissions of these files set to world or group writable, which is a bad thing. There were a few log files which I observed that could stand to have its permissions tightened up a bit. Also, if these files do not exist, you should make them via the "touch" command since the syslog will not make then on it's own.

  UTMP: This log lives in the "/var/adm/utmp" file and is a snapshot of the users who are currently logged into the system. It can be interrogated via the commands: "who, users, and finger". Look for anomalies, sudden reboots and unauthorized Su's.

WTMP: This log keeps a record of the users login and logout activity in the file "/var/adm/wtmp". Interrogate this file using the LAST command. It displays newest to oldest. Look for anomalies, sudden reboots.

MESSAGES: This log is a general repository for all sorts of syslog files. To figure out exactly what is in this log, you must interrogate the "/etc/syslog.conf" file. Most importantly, TCP trace, FTP, and TELNET logs will appear here. User errors and denied logins, along with rejected connections should appear here as well. Keep keen watch for these items.

AUTHLOG: This log stores all of the SU logging, and user logging event information. It's useful to see if anyone shady is attempting an unauthorized SU.

- Security Patches and Software Upgrades
  There have never been ANY security patches ever applied to this Solaris system and no system software upgrades. Fortunately, this Solaris system is running a more current release of SunOS 5.7. It is highly recommended that the latest patches from Sun's "sunsolve.sun.com" site are procured and applied. It is also highly recommended that Sun's Patch report be thoroughly interrogated for this particular version of the OS to ensure that no gaping security holes exist or have been recently discovered.

- MISC Issues
  Legal Messages: For reasons of legality, it is recommended that files "/etc/motd", "/etc/issue", "/etc/default/telnetd" be edited and the appropriate verbiage inserted to cover the legal bases should a compromise ever occur and the offending evil doers are apprehended. The "TELNETD" file should hold the command:

  BANNER="TEXT MESSAGE FOR BAD GUYS"

  The FTPD file in "/etc/defaults" behaves similarly and can also permit the UMASK to be set for uploaded files. There is a CRON file there as well which logs cron information.

  TCP Sequence Numbers: Also, the "/etc/default inetinit" file has the following variable in it which forces the system to use a better randomization algorithm for TCP sequence numbers, making hijacking attacks harder. However it is set to "1" and it needs to be set to "2". It is recommended to make that change.

  TCP_STRONG_ISS=1

  Passwords: The file "/etc/default/passwd" allows the parameters MAXWEEKS and PASSLENGTH to be set, effectively turning on password aging and tweeking the length of passwords.

  Keyboard: It is a possibility to disable the STOP-A from the keyboard by uncommenting the following variable:

  #KEYBOARD_ABORT=disable

  from the file "/etc/default/kbd". However, if the systems get hung up, only power-cycling will help. This is bad since there is no opportunity to run "SYNC" at the prom level.

- Host Based Vulnerability Detection
  At the risk of redundancy, it is critical to accent the need and the lack of, any host based vulnerability testing. None exist on this Solaris server, nor is there any evidence that any

have ever been run on it either. Programs like COPS, TIGER/TARA, LOGCHECK, LSOF, WATCHER, NESSUS, or NMAP are recommended to use to determine the extent and severity of weaknesses and vulnerabilities of the system.

- Host Based Intrusion Detection
  Intrusion detection and flexible reaction mechanisms are an absolutely critical layer of defense around the host. Again, there are no processes on this Solaris server, which can interrogate the packets coming in from the wire, examine them for legitimacy and take defensive action on that packet and it's offending port should the packet be considered questionable or if a genuine attack is detected. It is HIGHLY recommended that either SNORT or PortSentry be procured, implemented and put into service as a regular defender of this hosts security.

THIRD PARTY APPLICATION VULNERABILITY

Introduction
This section examines the vulnerabilities present in third party software such as web serves and CGI scripts. The vulnerabilities and securing against those are interrogated and discussed. When ever a third party software program is installed onto the system, new complexities arise and also new security hazards. It's the judicious process of minimizing these risks which is the focus of this section.

Summary
There are some weaknesses identified stemming from the third party software, particularly Apache configuration, Sendmail vintage vulnerabilities and CGI scripts which are questionable. These are fairly easy fixes in the grand scheme of things and should be done quickly.

Recommendations
It is recommended that the directives articulated in the "Apache Security Tips for Server Configuration" document get executed immediately, and that the vintage of Sendmail get replaced immediately. The CGI scripts concern does not seem to be a highest priority and is more if an investigative effort rather than an immediate fix of a clearly identifiable security hole.

Specific Critique

- Apache Web Server
  The apache server was installed and compiled with Open_SSL and MOD_SSL so there appears to be a secure environment for performing HTTPD communications. Upon further inspection, none of the directives articulated in the "Apache Security Tips for Server Configuration" document have been executed, so things like directory listings are possible. It is an absolute requirement that the security directives from Apache be implemented ASAP. There is also a configuration problem with the SSL authentication at the moment which prevents the SSL module from working at all. Fixing that is another high priority.

- CGI-Scripts
  A test BBS script taken from Extropia.Com worked fine but upon further inspection produced the following world writable files, which should be inspected for their security implications thoroughly. CGI scripts in general present one of the biggest security threats to an otherwise healthy server and made it to the SANS top 10 vulnerability list.

  World-Writable Files:
  find /usr/local/apache/ -type f –perm –o+w –print

/usr/local/apache/cgi-bin/BBS/Msg_Open/000001-000000.msg
/usr/local/apache/cgi-bin/BBS/Msg_Open/000002-000001.msg
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e5db05202428.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e6090523cdd2.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e65405270ddd.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e6b5052c0397.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e6d6052e90b8.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e6eb0531012f.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e8340544ea29.dat
/usr/local/apache/cgi-bin/BBS/Sessions/3a16e8b8059e66c3.dat

- Sendmail
  This system is running Sendmail and Pine as the MTA and mail reader. The Sendmail vintage currently running is the stock vintage running with Sun Microsystems Inc., SunOS 5.7 Generic October 1998 release, which can't be all that secure. It is highly recommended that it be completely shutdown until a more updated version can be procured, implemented and tested. Sendmail also made it to the SANS top 10 vulnerability list. Along with the discussion of Sendmail is Qpopper, which is very much out of date and has shown up on the BugTrac listings recently. It should be updated immediately.

SANS TOP 10 VULNERABILITY LIST
(Excluding Windows)

Introduction
This section summarizes the targeted vulnerability assessment of the SANS top 10 list of critical Internet security threats. It's purpose here is to see where this Solaris server stands with regard to the threats associated with each identified vulnerability.

Summary
This Solaris server isn't stacking up very well to the standard of vulnerability that SANS has set forth. This server has gotten bad reviews in every category and the actions needed to correct these problems should be taken immediately.

Recommendations
The directives articulated in the SANS top 10 Vulnerabilities document describe in detail how to eliminate the below internet security threats. It is recommended that those steps should be taken very soon.

Specific Critique

- Bind
  This server is using BIND vintage 9.x. It's a complete rewrite of the base BIND codebase and you know what that means. The good thing is that all of the old vulnerabilities were fixed! ;-)

- CGI-Programs
  The current CGI script base is very small, but it will grow. The most common source for CGI scripts is Extropia.Com. An examination of file permissions have already turned up

world-writable files. Although this doesn't seem to be a real concern right now, their particular brand of scripting should be kept an eye on.

- Remote Procedure Call
  There should be NO remote procedure calls ever on this server, however the out of the box installation has NFS running. This is a problem.

- Sendmail
  The Sendmail vintage is OLD and probably insecure. It represents a security hazard and should be upgraded.

- Sadmind and Mountd
  These processes are indeed running on this Solaris box and should be removed as they represent a serious security threat.

- User ID's and Passwords
  Some password security measures have been making progress, but there is a need for an initiative to enforced the use stronger passwords. Anonymous and guest accounts have been removed. Steps can be taken to mitigate the ".rhosts" vulnerability.

- IMAP and POP
  Qpopper does exist on this system and is very much out of date. Since it has shown up on the BugTrac listings recently, it should be updated immediately.

- SNMP
  SNMP is running on this server and should not be. It is a security hazard and should be disabled from the startup routines.

REFERENCES AND CREDITS

Spitzner, Lance, Armoring Solaris
http://www.enteract.com/~lspitz/armoring.html

FreeBSD Security How-To
http://people.FreeBSD.org/~jkb/howto.html

Security Tips for Server Configuration
http://spiderman.HPI-INC.NET/manual/misc/security_tips.html

Security report examines BeOS and BONE
http://www.beosjournal.com/BIAbeos_security.shtml

How To Eliminate The Ten Most Critical Internet Security Threats
http://www.sans.org/topten.htm

Help Defeat Denial of Service Attacks: Step-by-Step
http://www.sans.org/dosstep/index.htm

Setting up a basic DNS server for a domain
http://www.ludd.luth.se/~kavli/BIND-FAQ.html

Spitzner, Lance, Know Your Enemy Series
http://www.enteract.com/~lspitz/enemy.html
http://www.enteract.com/~lspitz/enemy2.html
http://www.enteract.com/~lspitz/enemy3.html
http://www.enteract.com/~lspitz/forensics.html
http://www.enteract.com/~lspitz/motives/
http://www.enteract.com/~lspitz/honeypot.html

Spitzner, Lance, Armoring Linux
http://www.enteract.com/~lspitz/linux.html

Spitzner, Lance, Armoring NT
http://www.enteract.com/~lspitz/nt.html

Spitzner, Lance, Intrusion Detection
http://www.enteract.com/~lspitz/ids.html

Wietse Venema : Internet Security Auditing Class Handouts
http://www.porcupine.org/auditing/

Freeware for Solaris
http://carroll.cac.psu.edu/pub/solaris/freeware-www/
Tripwire
http://www.tripwiresecurity.com/

Snort
http://www.snort.org/

Secure Shell
http://www.ssh.org/

Frequently Asked Questions about BIND
http://www.nominum.com/resources/faqs/bind-faq.html

Last Updated: May 23rd, 2017

Upcoming Training

SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**