Global Information Assurance Certification Paper

GIAC LEVELTWO

PRACTICAL ASSIGNMENT

SANS DEC 2000, Monterey

Conference Track 6

SECURING UNIX

Mukul Agarwal

Table of Contents

Operating System Vulnerabilities

1. **OS Installation**
2. **Patch Administration**
3. **Service Administration**
4. **File System Configuration**
5. **Logging**
6. **Hardware Administration**

Configuration Vulnerabilities

Risks from installed third-party software

1. **Apache (Web Server)**
2. **Sendmail (Mail server)**
3. **Bind (DNS Server)**

Administrative Practices

Security patches up to date

Sensitive data is stored encrypted and how

Data is sent over the Internet encrypted

Backups/Disaster Recovery

Incident Response Plan

Disaster Recovery Plan

Physical Security

Prioritized List of Security Vulnerabilities and Issues

Prioritized List of Recommended Fixes

References

Executive Summary:

The company that is being audited is a major provider of security and validation products and services hence it tries/wants to maintain very high standards for security at both the

system and network level.
Facts: the company has standardized on Solaris as the Operating system of choice.
Applications: The Company is running BIND for DNS, Apache for Web server, Sendmail as the mail transfer agent and an ftp server, which is one of their products.
Audit: Though the audit was to look into the security structure of the entire company it was concentrated on the systems that were on the DMZ mainly web server, ssh-gateway/ftp server, and DNS server. The idea was to find vulnerabilities and suggest improvements.
The Following issues were found in the audit and later in the chapters I have dealt in detail as to what consequences it can have and what tasks can be done to rectify these issues.

There were Operating System Vulnerabilities in the areas of OS Installation, Patch Administration, Service Administration**,** file System Configuration, Logging, Hardware Administration which included issues like:

- Loading the complete OS packages on all systems for ease of use instead of what is needed.
- Not keeping the systems current with loading the latest patches on them.
- Running services that were not needed and modifying the boot up scripts and kernel parameters to make the system more secure.
- All sorts of configuration issues with passwords, user accounts, core files, NFS services, RPC services etc
- Lack of proper hardware support for critical systems. Ex: No 24/7-support contract, none of the critical systems running on UPS.
- There was no policies defined for incident handling incase there was a break in, No formal policy for disaster recovery incase of an earthquake, fire etc. Nor was their any formal policy regarding backups.
- Their is lack of Physical security for servers residing in a data center within a building, which is not manned by a security guard 24/7 nor had any video cameras on each of the doors.
- One of the most disturbing facts was the lack of proper logs being maintained and worst of all no concept of periodic log checking.
- There is no Intrusion Detection System in place for Vulnerability Scanning.

I have dealt with all the above issues in detail in next few chapters and have also included the list of tools that I think are important for any site to evaluate their security.

Overview of Existing Infrastructure

Network
The network is run on 100BaseT. The network primarily consists of numerous hubs plugged into a single switch. Internet connectivity is gained through a Cisco router and a checkpoint firewall. Remote dial in connections are provided through the use of

checkpoint VPN or SSH

Systems
The assessed systems are all running Solaris OS 7.

Operating System Vulnerabilities

OS Installation: The OS installed on the assessed systems were installed with full OEM software packages plus support thus opening unnecessary vulnerabilities associated with packages, which are installed but are not needed.
Also in general different systems were installed with different configurations in terms of OS, patch levels and mounts making it difficult to manage.
Recommendation

- Do not install the largest and the most complete installation package instead download the smallest bare minimum operating system image provided by Solaris, which meets the requirements. Ex: install the "Core System Support" Image for the Internet connected platforms.

Patch Administration: Recommended and security patch cluster updates were not kept up to date causing the system to be exposed to well documented vulnerabilities at a later time.
Recommendation

- Use the Sun's PatchDiag tool, to discover the outdated patches on the different hosts.
- Can try to automate the patch update process by use of patchdiag.xref file where this file is downloaded on a regular basis and compared with the patches that are already installed on the system. If any systems are found lacking, the needed patches are downloaded automatically from http://sunsolve.sun.com and installed on the system in question.

Service Administration: Lot of unwanted services were left running on the system that were not needed. This just increased the risk of vulnerabilities associated with each of them. A number of *.rhosts*, *.netrc* files and */etc/hosts.equiv* have been defined for trust relationships for users and machines with other machines in the internal network which can cause a breach in security.
Recommendation

- The need for each service should be reviewed on each server, and all unnecessary services should be disabled by:
- Commenting out the service in */etc/inetd.conf* so that the service is not started at boot
- Removing the unnecessary system binaries from the /etc/rc.d directories

- Renaming the start scripts in /etc/rc#.d directories from S to s if you are not a fan of removing the start up scripts.
- NFS and RPC based services perform limited authentication and are a significant risk. Depending on the level of security versus convenience factor should be modified or removed.
- The need for the 'r' services should be reviewed, and disabled wherever not necessary. Alternate and more secure methods of providing the same functions to a user should be incorporated (e.g., *telnet* with SSH instead of *rlogin*).
- Users should not be allowed to establish trust relationships. The system should be swept for *.rhosts* and *.netrc* files daily, and users should be educated about the security risks associated with their use. Global trust relationships defined through */etc/hosts.equiv* should be qualified and restricted to specific hosts as follows:
    1. Deny access to all hosts through */etc/hosts.deny* and then,
    2. Define specific hosts and users that can use the trust relationship in */etc/hosts.allow*
- Use a port-based access control method by implementing *tcp-wrappers* to specify which users and machines have access to a given service on a given machine
- Further, all remote access, especially for administrative purposes should use some end-to-end encryption and strong authentication technology like SSH.
- The network parameters are set in the /etc/init.d/inetinit file. There are many configuration steps needed there. These changes help stop many network based hacker attacks, such as "SYN-flooding" and "ARP-spoofing attacks." The machine should have /etc/notrouter file to enable static routing, which is the most secure. Lastly an alteration of the /etc/init.d/inetsvc file, to disable DHCP, multicast routing, and inetd services should be done. Removing inetd would automatically unable telnet, rlogin and thus access to the machine over the network unless sshd is installed

File System Configuration: The system in question did not have a separate /usr partition and also all the file systems were mounted rw, setuid making it vulnerable for the OS binaries to be replaced by Trojan horse programs and the danger of rogue set-UID programs being placed in any of these file systems.

Recommendation

- Set all the file systems to nosuid. The nosuid options tells the system to ignore any files that have a setuid permission and to ro options tells the system that the directory is to be set read-only. Setuid files are dangerous at all levels but they can't all be removed. Some are required for system operations. In addition the nosuid option also means nodev or no devices. That means that the nosuid option cannot be used on and system directory containing devices but directories other than the root file system can be set nosuid.

Logging: The system has only default logging, initialized by Solaris install that is turned on. Although, numbers of logs are generated locally, there is no formal process for

reviewing the logs periodically
In the absence of a periodic review of the logs, unauthorized activity may go undetected for a very long time until substantial damage has already been done to the system.
Recommendation

- The following logs should be created and maintained on each server:
  - /var/log/wtmp: login/logout history
  - /var/log/btmp: unsuccessful login history
  - /var/log/messages: syslog messages
  - /var/log/secure: access and authentication log
  - /var/adm/loginlog: to capture failed logins
  - /var/adm/sulog: su attempts log
  - /var/log/authlog
- The audit logs should be reviewed periodically, preferably every day. To facilitate the task, various freely available tools like *swatch* or *logcheck* can be used to identify traffic and access patterns, and flag suspicious activity before it escalates into an intrusion.
- Create a central logging server. Putting all systems logs onto a separate computer, inside the protected network, helps to insure the validity of those logs
- /etc/syslog.conf file should be updated to include "auth.info /var/log/authlog".
- Create a script to rotate logs on a regular basis.

Hardware Administration: There is No hardware redundancy if the CPU or the disk storage unit fails. Moreover there is no on-site hardware support and on-line tech support beyond the normal working hours, which has the potential for extended system downtime in case the hardware fails.
Recommendation

- For all the critical machines have a 24/7 support policy which includes software and hardware support
- Try mirroring the root drive to a secondary drive so that incase the root drive is corrupted or has a hardware failure you could boot the system from the secondary drive.
- Have some spare parts onsite incase of emergency.
- Once the hardware gets old try to replace the hardware by putting in a better hardware for machines that are running critical applications rather than waiting for a disaster to happen. Bottom line be proactive.

Configuration Vulnerabilities

1. root logins not restricted to console.
   Fix: update /etc/etc/default/login file to deny root logins except from console.

2. /etc/passwd file was not updated and still had old accounts. Few accounts had logins disabled but accounts still have a valid shell. Several user accounts that had been disabled were found to still have valid shells assigned to them.
   Fix: Although the account is disabled by altering the account's entry in the /etc/passwd field, an extra layer of protection against its unauthorized use can be implemented by assigning it a non-shell, such as /bin/false.

3. User Id's with very weak passwords. Running crack and "Joe-the-ripper" we were able to crack 90% of the passwords.
   Fix: Need a password policy should be in place which defines how the rules for password creation should be.

4. No restrictions on the generation of core files. On machines with a lot of ram a hacker-forced core dump can act like a denial-of-service attack, also a hacker can find restricted information in a core file that can be used to attack the system in other ways.
   Fix: This can be easily fixed by adding a parameter "set sys:coredumpsize=0" inside /etc/system file

5. Did not enable stack protection to prevent now infamous "buffer overflow" attack which causes the program to crash and and give the hacker a login shell with permissions of root.
   Fix: Add another parameter under /etc/system file "set noexec_user_stack=1"

6. Had some improper file and directory permissions.
   ***Fix*:** this can be fixed easily by downloading the program called fix-modes" by Casper Dik. This program can correct permissions on the files and directories by removing group and world write permissions. Yassp and Titan are two other programs that help to lock down a system.

7. Did not enable eeprom password security i.e. system does not require password to issue EEPROM commands.
   Fix: A major hole exists when there is no password assigned at all. An intruder with root privileges can assigned an unknown password using "eeprom security-passwd=" command and reboot. The machine will be unusable because administrators will be unable to boot without that password. The only way to recover from this is to install a new eeprom and can take weeks and requires vendor's assistance.

8. Identifying banners provide information on the server operating system, service version etc. These help an attacker identify the system and focus on exploits that can be used against the specific system, making it much easier to successfully break into the system very quickly.
   Fix: Identifying banners should be removed and replaced with generic banners, which

warn intruders against unauthorized access

Risks from installed Third-Party Software

The assessed machines contain one or the other third-party software to perform their assigned functions (e.g. mail serving, web serving, DNS serving, FTP, SSH gateway etc) We will look at the following applications that have been installed between the machines that are being assessed. **Apache** is deployed for web serving; **Sendmail** is deployed for mail services; and **Bind**, for name server. We will primarily focus on Apache, Sendmail, and Bind.

Apache (www.apache.org)

- Version based vulnerabilities
- In the httpd.conf file we're making sure that secure directories require logins.
- Don't allow harmful options are not allowed ex: (FollowSymLinks, ExecCGI, Indexes, AllowOverride, etc.).
- Setting the default access to deny should be the default. This will protect certain directories from unwanted users.
- Typical attacks on web servers are CGI script attacks and denial of service attacks thus make sure the directory permissions are proper
- Lock the whole directory structure with chroot.

Sendmail (www.sendmail.org)

- Sendmail binary was the one provided by Sun which is not as secure as the Open source version available from www.sendmail.org
- Sendmail was running as a daemon, which exposes the system to Sendmail vulnerabilities.
- Anyone can telnet to the Sendmail port 25 and get the information about the Sendmail version that was running on the machine. This would make it very easy for him to find out the exploits associated with this version of Sendmail.

Recommendation

For Clients

- Machines that act as a mail client should not have Sendmail running on them as a daemon but in fact should have sendmail run from cron every few minutes to flush the mail queue.
  Ex: 0 * * * * /usr/lib/sendmail -q

For Mail Servers

- Use open source sendmail version available from www.sendmail.org as it is more secure.
- Change the sendmail banner to something less informative for people who telnet to port 25 to see if they can exploit any sendmail vulnerabilities.

Bind (http://www.isc.org/products/BIND/)

- Typical Bind attacks are buffer-overflows and denial of service attacks.
- Was running an old version, which had vulnerabilities, associated with it. The bind version was upgraded to the latest.
- Split horizon DNS was implemented where there is an internal DNS server and an external DNS server.
- Run BIND from a chroot()'d environment. The DNS servers that are on this network are currently running BIND from root (/). BIND provides a mechanism where it can be run from a chroot'd environment. If an intruder is able to get a shell by compromising the named daemon, the amount of damage that can be done should be confined to only that special root directory.
- Hide the version of BIND through the proper configuration of the /etc/named.conf file

Administrative Practices

Most of the key policies are currently not defined and are being worked on. Right now there are:

- No policies in place for passwords management (aging, length, etc.)
- No Incident handling/Response Policy
- No Disaster Recovery Policy
- Servers Not Configured by Role
- No Intrusion Detection System or Vulnerability Scanning.
- Lack of Fire Suppression System
- No power outage handling

Without these types of policies being in place and known by system administrators causes a panic driven reaction if any of the above issues arises. It also makes system administrators make their own judgment when a situation arises and thus creates a high level of vulnerability to the infrastructure as a whole.

Security patches up to date

The servers that were assessed did not have the latest security or the recommended patches updated on them on a periodic basis. In fact patches that were added to them were from the time the machine was build and put into production which is anywhere from

This is a major security hole since with time as vulnerabilities arise it makes the machine an easy target for hacker community to break into.

Sensitive data is stored encrypted and how

The company does not encrypt sensitive data that is stored within the corporate walls. Sensitive data just resides under the user account that created it. If someone can login as the user then he would have the access to the sensitive information. If the user chooses to he could further password protect individual files he chooses to but again there is no formal policy to this effect and the decision is left to the user.

Data is sent over the Internet encrypted

The company has a number of remote offices, which are connected to the corporate office through T1's. Information is transferred between the offices encrypted with the use of IPSEC build into the Cisco router**.**
Employees logging from home or from the Internet connect using a VPN connection, which is encrypted using IPSEC**.**

Backup Policy

The company does not have a formal or documented backup policy currently in place. The company has had problems with its backups in the past and due to its spurious growth in such a short time, it is currently revamping their backups which includes getting new software, new tape libraries, new system etc.
Recommendation: Once all issues with the backups are resolved they should plan on a policy that would clearly define the following:

- What data is backed up
- How often data is backed up and the type of backup (full, differential)
- When the backups are scheduled and how they are verified
- How the backup media is handled and labeled
- How the backup media is stored
- How long the backup media is retained
- How backup media is rotated and expired
- How backup data is recovered

Having accurate backups is a critical factor for every IT department. The department should make sure things are in place and the policies are clearly defined.

Incident Response Plan

An incident is defined as a breach in security, whether the breach occurs externally or from within the organization. The presence of an incident response plan will assist in the identification and containment of an incident. Without a plan in place containment may occur slowly and evidence may be lost or corrupted prior to areas of authority being properly assigned.

***Recommendation*:**

- Create an incident response plan for the company**.** This will assist with the proper handling of data, containment procedures, recovery procedures, and how best to proceed with possible prosecution.
- Get a "Rootkit" with necessary tools that the can fit onto a CD to evaluate potential break-in
- Look into web sites for obtaining information on the tasks to perform if they are broken into: http://www.fish.com/forensics/, http://www.porcupine.org/forensics/, http://www.cert.org/, and ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist.

Disaster Recovery Plan

A disaster is defined as an event that can be very damaging to the company and can include the loss of a system, loss of power, natural disaster, or any other event that may result in a diminished IT posture. The presence of a disaster recovery plan will lend structure to the overall IT posture and provide steps that should be taken in the event of a disaster.

***Recommendation*:**

- Create a disaster recovery plan so that in the even of a disaster steps and procedures are available to the recovery team members. This plan will assist in defining disaster criticality, assigning areas of responsibility to recovery team members, and reducing down time in the event of a disaster.
- The disaster recovery plan should include areas of responsibility for disaster recovery team members, how usable hardware can be redistributed, the priority of services that must be re-established, a time-line of recovery events, and how to deal with the public (press, law enforcement authorities, etc.).

Physical Security

Servers are located in a secured lab within the building.
The data center is located behind an additional electronic cipher-lock to ensure controlled access.
ID badges are required for entry into the building.
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.

Recommendations:

- All entrances should be monitored by video camera
- There should be security personnel on duty 24 hours a day, seven days a week if the data housed in the building is of critical nature.
- There are no UPS (un-interruptible power supplies) incase of a power outage.
- Network is subject to unauthorized internal access or tampering from unattended ports

Prioritized List of Security Vulnerabilities and Issues

1. All systems installed with full OEM software packages plus support
2. Security and Recommended patches not kept up to date on systems
3. Issues with service administration. Unwanted services that are not needed left running on the system.
4. File systems mounted rw.
5. Limited logging, no periodic log checks and no centralized log server.
6. No 24/7 hardware/software support
7. Configuration issues pertaining to password policies, weak passwords, core file generation, file/directory permissions, buffer overflow issues
8. Root access from the net should be denied.
9. NFS should not be turned on or installed.
10. No PROM password.
11. RPC based services should be replaced by ssh based services
12. Third party software issues for software like Apache, Sendmail, Bind
13. Lack of policies regarding Disaster recovery, Incident handling, Backup policy
14. Lack of proper physical security

Prioritized List of Recommended Courses of Action

1. Install a system with smallest bare minimum operating system image
2. Maintaining Patches. Use a PatchDiag tool to keep your system up-to-date with latest patches.
3. Replace the default /etc/init.d/inetsvc with a minimized version containing only those commands required for the configuration of the network interfaces only.
4. Reconfigure machines for only required accounts, services, run levels, and processes
5. Turn off asynchronous PPP (/etc/rc2.d/S47asppp), auto shutdown features (/etc/rc2.d/S85power), wbem (/etc/rc2.d/S90webm), preservation of vi file edits if vi or session crashes (/etc/rc2.d/S80PRESERVE), Service Locater Protocol support

(/etc/rc2.d/S72slpd), uucp protocol support (/etc/rc2.d/S70uucp)

6. TCP_wrappers should be used to control access to the commands that must be in the inetd.conf file.
7. Force NFS clients to use privileged ports – for protection against some common script-kiddie NFS exploits

   Set nfssrv: nfs_portmon = 1

   Set nfs: nfs_portmon = 1
8. Do extensive logging and keep logs on a centralized log server.
9. Improve monitoring of both network and systems using tools like tiger, lsof, and nmap
10. Always have hardware and software support for all critical servers.
11. Try to minimize configuration vulnerabilities related to password, core generation buffer overflow, file and directory permissions etc
12. Always have prom password for all your servers.
13. Remove identifying banners and replace them with generic banners wherever possible.
14. Limit the number of third party applications that you run and if you do always make sure to constantly check their sites for vulnerabilities and updates.
15. Write and test a Backup Policy
16. Write and test an Incident Response Policy
17. Get a "Rootkit" with necessary tools that the can fit onto a CD to evaluate potential break-in.
18. Write and test a Disaster Recovery Policy
19. Evaluate and deploy an Intrusion Detection System and implement policies for Vulnerability Scanning
20. Beef up your physical security
21. Put all your critical servers un UPS

Prioritized List of Recommended Tools for Audit

COPS (use Kuang only to try and break into system) –
ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/cops

TIGER (use all of it to try and find potential problems on system)
–ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/tiger

Crack (used for cracking user's passwords – testing for strength of passwords) -
ftp://coast.cs.purdue.edu/pub/tools/unix/pwdutils/crack

Fix-modes – written by Casper Dik of Sun to find poor permissions on OS and alert and/or correct them.
ftp://coast.cs.purdue.edu/pub/tools/unix/sysutils/fix-modes

ISS – (internet security scanner). The free version has much less features that

commercially available version.
ftp://coast.cs.purdue.edu/pub/tools/unix/scanners/iis

Logcheck – used on syslog server to analyze log files for errors, or problems that admins should know about and send an email with info.
ftp://coast.cs.purdue.edu/pub/tools/unix/logutils/logcheck

Lsof – tool used to analyze what files are open and what processes are using them. ftp://coast.cs.purdue.edu/pub/tools/unix/sysutils/lsof

Nessus – awesome state-of-the-art scanning tool to find weaknesses in systems. Many DDOS attacks!
ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/nessus

Nfsbug – tool used to find nfs/ portmapper related weaknesses in systems.
ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/nfsbug

Nmap – tool used to map out the network contents, hostnames, rlogin hops, etc.
ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/nmap

Openssh – multipurpose tool used to encrypt basically all kinds of tcp traffic between machines, in lieu of telnet, rsh, rlogin, etc…
ftp://coast.cs.purdue.edu/pub/tools/unix/sysutils/openssh

Satan – a popular scanning tool (created by Farmer and Wietz)
ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/satan that has been superceded by:

Sara – scanning tool implemented by many gov. facilities to look for potential openings on systems. ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/sara

Saint – another scanning tool very closely based on Satan, except more modernized. ftp://coast.cs.purdue.edu/pub/tools/unix/netutils/saint

Shadow – a cool and powerful intrusion detection tool.
ftp://coast.cs.purdue.edu/pub/tools/unix/ids/shadow

Snort – slightly less feature IDS tool, that has been the Unix admins. Best friend for years. ftp://coast.cs.purdue.edu/pub/tools/unix/ids/snort

References

- "Solaris Security," by Peter H. Gregory. Sun Microsystems Press. Prentice Hall, Inc., 2000.

- http://www.sans.org/y2k/practical/Mandar_Rege.doc
- "Solaris Security Step By Step," by The SANS Institute. V. 1.0. 1999.
- Unix System Administration Handbook,
- Nemeth et al, Prentice Hall, ISBN 0-13-151051-7
- Building Internet Firewall, Chapman and Zwicky, O'Reilly and Associates, ISBN 1-56592-124-0
- "Solaris Practicum, Track 6: Securing Unix Systems," by Hal Pomeranz, Deer Run Associates. 2000.
- "Running UNIX Applications Securely, Track 6: Securing Unix Systems," by Lee Brotzman and Hal Pomeranz. 2000.
- http://www.sans.org/y2k/practical/Daniel_Robb.doc
- "Common Issues and Vulnerabilities in UNIX Security, Track 6: Securing Unix Systems," by Hal Pomeranz, Deer Run Associates. 2000.
- "Deploying DNS and Sendmail: SANS 2000 Course Book By Hal Pomeranz.
- UNIX@Night, UNIX Forensics, by John Green. SANS 2000 Course Book.
- http://www.sans.org/y2k/practical/John_Mcclure.doc.
- http://www.sans.org/y2k/practical/Tyrone_Lomeli_GCUX.doc
- http://www.sans.org/y2k/practical/Paul_Leadingham.doc