Practical Paper for the Unix Security Course (SANS 2000)
Monterey, California (15 Oct 2000 – 22 Oct 2000)

Proposing a security modem for an existing ISP

Executive Summary :

ABC is an Internet Company whose main operation is to provide Internet service to any customers who subscribe to its service.

ABC started in the 1998 with only 1 chief engineer and 4 engineering assistants. The services that are providing to customer would only be Internet access. There will not be other service. Assuming that the network connectivity and network security is well taken care of. We will be concerned and concentrating on the system security for all the existing services, DNS service, DHCP service, FTP and mail service.

Background

System Security on any operating system needs to analyze and examine before deployment. Subscribers will have better confidence in the service that we are providing if we have a good system. Services that we looked into in the existing system :-

(a) DHCP Service
(b) SMTP Service
(c) FTP Service
(d) DNS Service

We will be looking into the various ways of connecting the server like NFS. Samba, shell accounts and others.

After first analyze stage, we will focus into

(a) Defining the security policies of each individual systems

User
It would be useful if we are able to know each individual user who will be assessing the machines. These policies will be enforced by the System Administrator of each individual owners of the service machine. The owner should be aware of who is assessing the machines and what are the permission for each individual users.

Service/System

Servers should be configured to run as few services as possible, or just those services that is being offered in the server. With this, it will offer better administration to the system administrator in term of security, fault tolerance, scalability and reliability of the machine.

<u>Documentation</u>

Owners should need to develop a security document for future reference. This policy should be in written communication and should be recognized by your management so that it will be enforced throughout the ISP. However, without such policies and corresponding documentation, enforcement of the security document will be challenged legally.

(b) Vulnerability of the system

<u>Physically</u>

Equipment Inventory
If the inventory of the system is well managed, we will be able to track down all computer assets in the data center to prevent any theft or misplacement of the asset. If this is not being managed properly, machines might end up being "pick pocketed" by unethical employees and never being returned when they resigned or being terminated. If you want to take legal action, it will be not be that simple as there is no audit trail of the inventory hence making prosecution very difficult.
Action to be taken :
It had been a good practice to serialized asset tags on all equipment. With this serialization, we will be able to build database relating asset tag number to item description and system serial number. Individual component serial number will be given to equipment like memory, disk drive, tape drive, etc manufacture, date of creation, owner and so on.
These information will be useful for internal audit or to support new contract to the equipment. It will become a habit if all equipment is being tagged and enter into relevant data as soon as the equipment is delivered before deployment.

Locks
Electronic access control system should be deployed so as to track all incoming users to the system. With this system in place, any intruder will not be allowed into the premise. Or external vendors will be accompanied to the premises at all time.

In the next few section, I will be discussing at the various services offered by my company to end users.

Managing the Operating system in each server for various services

In this section, we will be discussing on to actions on the following items :-

(a) Setting up the operating system
   a. Identify the CPU architecture and initialize the system
   b. Calculate physical memory and initialize virtual memory systems
   c. Allocate memory for internal structures
   d. Configure system devices and initialize system resource
   e. Partition the Hard disk, mount the root file system
   f. Preparing the start-up script for each various Daemon on each server

(b) Configuring Starting the Network Services/File System

   Network Service
   a. Plan the Server diagram and IP address for each server
   b. Configure Network interface
   c. Performing Cron job to run process at various times of the day as desired

   File System
   a. Set up policy for system administrators to follow
      i. Knowing the directories structures
         i. Unix File System
         ii. Root File system
         iii. Physical File System

      Why need to know these file system ?
      ii. Knowing the File attribute, File Permission, User and Group Ownership
      iii. Knowing the Inodes and Data Blocks
      iv. Rotating the log files
      v. Knowing the Unix Privilege
         i. Understanding /etc/passwd, /etc/shadow to be able to deploy and manage user Ids well
         ii. Understanding the UIDs, Superuser and Groups
         iii. Understanding Chroot

(c) Unix Process running the Services
      i. Running those required services on specified machines
      ii. Killing unnecessarily daemons
      iii. Give priority resource to those required process

Action Plans to look into how to manage the service well

With the above section, we are looking at ways / resolutions to actually rectify the specified problem. In each sub-section, we will be discussing them in detail and hope to achieve the best of result.

Setting up the operating system

- Identify the CPU architecture and initialize the system
   a. DHCP Service
   b. SMTP Service
   c. FTP Service
   d. DNS Service

For any service, you would have to know the no. of users that you are supporting so as to make the right recommendation to protect the system as well as to ensure that the system is of optimal status with abundant/sufficient resources to support the security feature. We would also have to take into consideration the I/O as DHCP will always be the system that issues IP addresses to users who are requesting for it. Whereas FTP service, it would need abundant disk space so that users are able to store the database or any information into their respective directories. However, SMTP would also require disk space and disk I/O so that users can write and read their mail information.
Finally, the system will probe for device information, e.g disks, tapes, monitors, keyboards, etc.

- Calculate physical memory and initialize virtual memory systems
  After identifying the CPU architecture, the system will start to calculate how much memory is available on the system and initialize and allocate the virtual memory to each individual system, so we would have to ensure that we do not stress out the system. To ensure that the system is being fully utilized, unnecessarily daemon should be terminated to prevent them from virtually "eating" the resources.

- Allocate memory resource for system devices
  Once the system CPU and memory have been initialized, the kernel allocates memory for several internal structures. These structures will control the file system, keeping track of the processes on the system, and other critical kernel functions.
  They are namely devices attached to the system: disks, tapes, monitors, keyboard, etc.

- Partition the Hard disk, mount the root file system
  For each system, we would have to partition the harddisk carefully so that we can fully optimize the disk space and manage the service in non-root environment.
  a. DHCP Service (ISC)
     In our environment, we would prefer the DHCP Service to be run in chroot environment so that they will not be a threat to the system in any case of a hacking incident to remove important files like vmunix or any kernel files. The log files will have to be stored in another file directories as they are important in any case of the system is being hacked.
     All IP addresses that are issued are to be logged down for record purposes. Hence, we should consider this under another file domain.
     The recommended structure WILL BE discussed at a later stage.

  b. SMTP Service
     For the SMTP service, we would model it as the same structure as the DHCO service in this way, our machines will be consistent. We will not

run the service in root environment however, we will have the SMTP being ran as chroot environment.
Keeping the log files for each transaction will be a hesitate for any operators as the log size will be huge and hard to manage as it grows exponentially. However, the ISP's best recommendation is not to monitor where the users are accessing their pop account to download their mail users. But to be able limit the number of email send from a user to prevent spamming another subscribers' account.

c. FTP Service
For FTP services, the most preferred way to manage this is to run the service in chroot environment the same strategy applied.
As for security features in term of logging, it would be a good practice if all incoming and outgoing transaction be logged so that we are able to keep track of any hacking via FTP. To fully, manage the FTP daemon, we would also limit the file size in order to prevent over exerting the CPU intensively.

d. DNS Service
Iin the DNS Service environment, the service should be manageable in the chroot environment. The same explanation applied.
As for security features in term of logging, it would be a good practice if all resolved incoming and outgoing transaction is logged.

For all the abovementioned feature, we would assume that the syslog daemon is running in each system to enable logging on the ftp or ssh services.

- Preparing the start-up script for each various Daemon on each server
In all the various specified services, we will be using different start-up scripts so as to prevent unnecessarily service being ran on the machines. An example would be if you are running DHCP service, you would need the basic configurations are like inetd, syslogd and other telnetd. These services would allow you to manage the server remotely. Special Start-up scripts are required for each service as follows:-
   a. DHCP Service
   /etc/rc.local/dhcpd – to start the DHCP service automatically
   /log/ (under a different specified for storage) – housing of the log for DHCP to allow easier tracking and checking in any case of any hacking incident. We would also track the telnet session incoming traffic to the servers. It would be an ideal situation if we are able to track down the outgoing traffic from the server.
   b. SMTP Service (a separate server)
   /etc/rc.local/smtpd – to start the Mail Service automatically
   to have consistency across the platform, it would be good to have the same file directory so as to allow easy manage of the other server.

/log/ - will be used to house other logs and this will facilitate all logs to be stored in the system.

c. FTP Service
/etc/rc.local/ftpd – to start the FTP service automatically
the same applies here, we will use /log as the generic storage area for all servers.

d. DNS Service
/etc/rc.local/named – to start the Named service automatically
/log will be the storage area for the incoming and resolved results

Configuring Starting the Network Services/File System

Network Service

a. Plan the Server diagram and IP address for each server
In order to manage the scalability of the server area, we would have to plan for the IP range so that we would not have to change the IP address of the specified server again. With this IP range, there are ways to actually determine if the IP addresses are being used for a start.
    - look up the /etc/host file
    - network database lookup (if you have but since this is a new environment, you would not have to worry about IP collision
    - Domain Name Service

b. Configure Network interface
To configure the network interface of the Ethernet slot, it would be good if we could segment the IP from each individual range example,
x.x.23.1-7/255.255.255.248 for the range of DHCP Server farm
x.x.23.9-15/255.255.255.248 for the range of DHCP DNS farm
:
:

Performing Cron job to run process at various times of the day as desired

File System

Knowing the directories structures
The reason for us to know the various file system is to enable ourselves to

i. Unix File System
To fully utilize the file system, we would need to understand the limitation of the file system that is file should not have any null or special characters like "/". File names should not be longer than 255 characters. It does not support directory pathname longer than 1024 characters.
To better allocate the file storage area, we would need to understand how the structure of the Unix File System is being organized so that we would be able to concentrate on the root directory as well as privilege directory. These directories should be protected at all time and they should only be readable directory, namely :
/ - root file system, top of directory hierarcy
/dev, /devices – directory containing the system devices\
/usr – primary OS directory

These directory should be classified under the second categories which they can be used and view by privilege users
/var
/var/tmp
/usr/sbin/

The last type directory would be use by users,
/opt/
/usr/local
/home/
/export/home

ii. Root File system

Under the root directory, the components critical for system boot are located here. In particular the kernel itself resides near the top of the root file system, directly under /vmunix, which we need to secure this / directory as it had the kernel itself.

/sbin directory contain the INIT program and the program which INIT needs to configure the system. These programs include Unix command shell as well as startup script for the service, like DHCPd, SMTPd, Named.

/etc is where all the configuration file on the system resides. During the boot process, this directory is consider a read-write directory. Hence we would have to prevent "external" source from hacking into the system when it starts to boot up.

iii. Physical File system

Why need to know these file system ?
With the knowledge of the File attribute and File permission of each directory, we would be able to make classification on who should own the fileset and who would be able to access the files under the fileset. Also, we would be able to implement the User and Group Ownership of the server in term accountability.

For some cases, we would be able to use the sticky bit. The purpose of the sticky bit set was supposed to "stick around" in the memory of the operating system after the program had finished executing – this was a win on programs that needed to be executed constantly. We could also use UID and GID to control the logs files of each server to prevent unauthorized use of the log.

As mentioned before, we need to constantly monitor the log file, we would need to rotate them constantly so that we can keep these data "fresh". To do so, we would need to deploy the logrotate command to "refresh" all these logs. TO archive (gzip/compress) them, we would flush these log to the storage area for future reference.

(c) Common Issues/Problems that will be encountered if no action is being taken (including solutions to these vulnerabilities)

Some of the common vulnerabilities and solutions to them :-

a. Password
   Password is the key access to all machines. It should be well guarded in all ways. Also, we could deploy non-reusable authentication technology or a token that constantly change the pin. There are lots of password threats some examples are as follows:-
   i. Sniffing
      If the server is not equipped with SSH-3Des, it would be very vulnerable and any clear-text passwords can be "seen" from the network. Or, a computer can be configured to capture all data moving past it on the network. Potential attackers prone to install software which monitors all network traffic to obtain unencrypted password. The attackers would simply watch the network stream for a "password:" prompt and capture the response from the user.

      One of the best defenses against sniffers is to use a switched Ethernet network, each machine is only allowed to see the traffic on its leg of the switch. Also, this will provide us with a better network performance and a cheap and reliable solution. However, attackers can still install a sniffer on some central critical machines like our DHCP server, DNS server to capture passwords sent to that server, even if the server is on a switched network.

      Henceforth, to best overcome this, it would be best to install strong encryption technology program like Secure Shell or using VPN solution or both.

   ii. exhaustive guessing
      This method of exhaustive guessing is a slow, noisy way to try and break into a system. However, after 3 –5 unsuccessful attempts, the system will automatically drop the connection. And the system administrator will be able to view the log file for any attempts into the system => the usefulness of the log file.

To actually resolve this, it would be recommended to have the users to key-in at least one numeric character in any part of the password and the password must be 8 character long.

An improvement to Unix password security was to get the encrypted password strings out of the /etc/passwd file and put them into a file called shadows which is readable and accessible ONLY by root.

Other way to improve security would be to implement password expiry, password aging/history, and minimum password lengths with mixture of both numeric and text, force good passwords at change time.

b. Attacking running programs

   i. Core Files
   From time to time on a Unix platform, core dump will be generated. These files are the remains of a system process that has aborted unexpectedly and contain debugging information which can be useful in figuring out why the program die at any moment. In short, core files contain a complete imagine of memory allocated to the program at the time of crash. Just like the black box in the airplane. By decoding the strings program, many interesting information may be gleaned.

   Ways to prevent cores files from being generated:-

   (a) on a per-user basis using one of the command ulimit –c 0 configure in the .profile file where the system will not dump any core file in the users domain. However, user may still choose to re-enable core dumps if they wish.
   (b) on a system wide basis in the kernel which can be perform by most modern Unix system. It allows system administrator to easily set a kernel switch which prevents any process on the system from dumping core. This means that no user will be able to get a core file from software that may perceived to be a problem.

c. Chroot()
Chroot() is a unix system call that allows a process to give up access to all but a small portion of the file system. This would facilitate programmers to create processes which run in captive environment and therefore reduce many of the security risks.
WHY chroot() ?

One of the main uses for chroot() is to run networked services in captive environments so that security holes in these services can't be exploited against the entire machines. Typically, we will be using Chroot() to run all our services securely. Another reason would be the fact that chroot() is also useful for testing new services when you are not sure how secure a given service maybe. Chroot() is proven to be useful for creating captive shell accounts for users. With this setup, users are able to run only this application and to prevent people who steal their logins from doing damage to the rest of the system. Forcing these users to log into a chroot()ed shell is a good mechanism for limiting system access.
As some daemons have the functionality built-in to run in chroot() environment, like FTP and BIND, so there is no configuration needed and hence it would be rather very simple to manage it. One disadvantage is that chroot() can only be done by root.
There are questions that surface on why no chroot() everything since it is a fabulous idea. However, chroot() itself do has its limitation that is it does not allow a program which access many other files to be chroot()ed. These programs are deeply intertwined in the operating system that it's effectively impossible to run them in chroot()ed environment.

d. Network Attacks
In network attacks, we could look at various techniques to discuss on this :
(a) .rhosts file
users are allowed to specified remote hosts.users which can rlogin/rsh without password which this poses a danger to security as there is no authentication required to access the system.
(b) Similar to .rhosts. we could look at /etc/hosts.equiv. Hosts.equiv specifies trust relationships which apply to all accounts on the system. However, this may not be an excellent techniques though it had reduced some work. It still require no password ➔ security compromise.
(c) IP/Host based authentication
In addition to these .rhost and /hosts.equiv, we could deploy IP/host-based authentication to make the system more secure. However, this may not be a good idea as IP address can be spooled, DNS entries maybe corrupted, Root privs on other host can be exploited. Even for reversed DNS entries, it may also be spoof.
(d) As far as possible, in our environment, it would not be a good idea to manage .Xauthority as it poses the same problem as listed above.
To count-strike at network attacks, we can fully utilize the technology to deploy equipment like
1. use firewall to block access/spool address
2. make .;rhost/hosts.equiv mode 600
3. regularly remove .rhosts files
4. disable .rhosts in libc
5. Don't run rlogin/rsh but use SSH

6. Blocking common X ports at firewall
7. Use SSH to tunnel remote X events with kerberos authentication

Why SSH ?

SSH gives you better encryption as well as it could prevent eavesdropping and sniffer attacks.

(d) Resolution to problem

Unix Process running the Services securely
Running those required services on specified machines
Killing unnecessarily daemons
Give priority resource to those required process

Syslogd runs as a daemon and accepts messages from local program. It is started at boot time and usually creates two different communications channels. Syslogd listens to port 514/udp for message from other systems, it is able to send its own messages to other systems and forwarding messages from one system to another. However, the syslogd does not have any form of authentication, so one denial of service attack will fill up a system's logging partition with bogus messages. This hackers can hide their real transaction from the server operators. In order to curb this, it would be good if you could block using a firewall the external connection on 514/UDP. Different message useful message provided by the message facilities like Kern where is will log system kernel message which are important for troubleshooting on the failure of the server. Daemon facility is used by other system daemons (named, NTP, Etc) except for cron which has its own cron facility. Under the priority level, the syslog is able to classify them so that it would be manageable by the server and look at the severity of the problem to solve it, for instance, if emerg appears in the log, it would have to be resolved first else the rest of the other services will be affected.

In the FTP server, since there are lots of known vulnerabilities, it would be advisable to continually upgrade the daemon for the service that we are providing. It would be good if we could constantly check for new update on the security loophole of FTP service.

In operating the FTP service, it would be good if we ran it in chroot() environment and log all session in syslog (as above) as well as to control the access using /etc/ftpaccess or TCP wrapper. To tighten security, it would be good if guest account/anonftp account be removed. For FTP directory, these would be the recommended /bin and /etc/UID/GID root mode 111 (execute only) and /pub UID root, GID ftp, modem 02555. Using TCP.wrapper to limit access to local domain and one other network. Should constantly monitor /var/log/secure for connection, /var/log/xferlog for file transfers.

Common security issues in BIND :

A. giving away too much information
   this will be an avenue that attackers will probe the system and attack after gotten the relevant information.
B. buffer overflow
   - Root compromise attacks
   - Denial of service type attacks
C. cache poisoning
   Cache poisoning occurs when a name server had been tricked into believing errorneous information from some external source.

Ways to counter-act the attackers:-

a. Bind V8 allows named to run without superuser privileges
b. Bind V8 can also be run in chroot()ed environment
c. Fine-tuning the configuration setting will help protect against buffer overruns and other compromise attacks
d. More difficult setup and management

In the appendix would be the checklist done for a generic system in term of security features. However, it would be good if we could explore in-depth into various protection mechanism like the following:

(e) Future Protection
   a. Installation of Firewall
   b. Intrusion Detection System
   c. Monitoring Tools to be used for monitoring the system

Thank you.

Reference :

Securing Unix System