A REVIEW OF METHODS FOR SECURING LINUX OPERATING SYSTEM

[1]V.A.Injamuri
Govt. College of Engineering,Aurangabad, India
[1]Shri.injamuri@gmail.com

Abstract— This paper is focused on practical securing Linux production systems. It discusses basic Linux Security requirements for systems that need to pass various audits in an enterprise environment. This Linux Security is intended for a technical audience, Linux system administrators, and security people in corporations and organizations that have to use commercial Linux distributions for their production environment

***Index Terms*— iptables,RPM,inittab , boot script**

INTRODUCTION

There is a need to make Linux production systems compliant with various audit requirements; the system can offer a good baseline and starting point. The main objective of the system is to discuss basic Linux security requirements including account policies for production systems that are being audited [1].

Physical Security

Physical security should be of the utmost concern. Linux production servers should be in locked datacenters where only people with passed security checks have access. But physical security is out of scope for this article.

Verifying Security Action Items

To improve security, there are scripts available which can verify that all security action items have been executed. Even the best sys admins can make mistakes and miss steps. In case of larger Linux environment, it would be a good investment to write scripts for checking Linux security action items.

Retiring Linux Servers with Sensitive Data

To retire servers with sensitive data, it is important to ensure that data cannot be recovered from the hard disks. To ensure that all traces of data are removed, the Disk Sanitizer tool can be used...

Backups

In the event of the system being compromised, the backups become invaluable. In cases like bugs, accidents etc. backups can be used to compare you current system against your backed-up system. For production systems it is very important to take some Backups offsite for cases like disasters. For legal reasons, some firms and organizations must be careful about backing up too much information and holding it too long.

Disk Partitions

Servers should have separate partitions for at least /, /boot, /usr, /var, /tmp, and /home. It is not desirable to fill logging and temporary space under /var and /tmp using up space of all the root partition. Third party applications should be on separate file systems as well, e.g. under /opt.

Firewall (iptables)

The system will not cover iptablesmost companies use hardware based firewalls to protect their servers in a production network, which is strongly recommended for such environments.

Kernel Security Features

Virtual Address Space Randomization:

Starting with the 2.6.x kernel releases Linux now uses address-space randomization technique to mitigate buffer overflows.

SELinux

SELinux is an advanced technology for securing Linux systems. Hardening Linux using SELinux technology, on its own, warrants its own security[2] .

FTP, telnet, and rlogin (rsh)

FTP, telnet, and rlogin (rsh) are vulnerable to eavesdropping, which is one of the reasons why SSH/SCP/SFTP should be used.

I. PROBLEM STATEMENT

At the heart of Linux system is the Linux kernel and operating system. Combined, these form the base level of the system on which all the applications run. Comparatively speaking, the Linux operating system and kernel are actually reasonably secure. A large number of security features are built in the kernel, and a variety of security-related tools and features come with most distributions or are available in open-source form. Additionally, Linux offers exceptional control over whom, how, and what resources and applications users can access.

The security of the system depends on a wide variety of configuration elements both at the operating system level and at the application level [3].

Additionally, the Linux operating system and kernel are complex and not always easy to configure. In fact, Linux systems are nearly infinitely configurable, and subtle configuration changes can have significant security implications. Thus, some security exposures and vulnerabilities are not always immediately obvious, and a lack of understanding about the global impact of changing configuration elements can lead to inadvertent exposures. Furthermore, security on Linux systems never stays static. Once secured, the system does not perpetually stay secure. Indeed, the longer the system runs, the less secure it becomes. This can happen through operational or functional changes exposing the threats or through new exploits being discovered in packages and applications. Securing the system is an ongoing and living process.

Many distributions come prepackaged or preconfigured with a recommended default set of packages, applications, and settings. Usually this configuration is based on the author or vendor understanding what their end user requires of the distribution. Generally speaking, a lot of this preconfiguration is useful and enhances the potential security of the system; for example, Red Hat comes preconfigured to use Pluggable Authentication Modules (or PAM) for a variety of authentication processes. But sometimes this preconfiguration opens security holes or is poorly designed from a security perspective [4].

For example, as a result of the vendor's desire to make it easy to set the system up the vendors may install, configure, and start applications or services. Red Hat automatically configures and starts Send mail as part of the default installation options.

To be able to address different security issues, there is a need to have a solid understanding of the underlying basic security requirements of the system [5].

II. LITERATURE SURVEY

Thus we found that the linux security is centred on the how these security parameters are set and how configurations files are configured. Each server has its own configuration file and proper configuration of these files lead to good security of particular server. To achieve high security admin need to have very careful about configuring all security related configuration attributes and there high security attributes values[6][7][8].Thus high security is achieves through proper configuration of system, server and services configuration files and applying security related parameters. The summary of the vulnerabilities, attacks and defense mechanisms is given below:

Table 1: Vulnerabilities of workstation security and remedy

Sr No	Vulnerability	Attacks	Countermeasure
1.	No separate partition for /boot, /, /home, /tmp, and /var/tmp	System crash and data loss	Create separate partition for /boot, /, /home, /tmp, and /var/tmp
2.	Unnecessary software's	Software vulnerability attack	Install minimum software's
3.	maliciously altered package	System instability ,System crash and data loss, data still	Install Signed Packages
4.	No BIOS password	Stealing/Changing Data Using a Bootable Linux CD	Give BIOS password
5.	Single User Mode access	Access as root user without password	Password protecting BIOS
6.	Access to the GRUB Console	change its configuration or to gather information using the ***cat*** command.	Password protecting GRUB
7	Access to Insecure Operating Systems	If it is a dual-boot system, an attacker can select an operating system at boot time (for example, DOS)	Password protecting GRUB

Table 2: Network Security vulnerabilities and countermeasures

Sr No	Vulnerability	Attacks	Countermeasure
1.	OS fingerprinting	Get os information like OS version etc.	Place login banner
2.	Local log monitoring	Remove of log entries and log files	Remote log monitoring
3.	Insecure Services FTP , Telnet Transmit Usernames and Passwords Over a Network Unencrypted	1) Get user name and password. 2) Denial of Service Attacks (DoS)	1) Avoid these services and use behind the firewall 2) Use tcp_wrappers and xinetd 3) Use SSH
4.	/etc/sysctl.conf configuration file vulnerability	1) SYN Attack 2) IP Source Routing 3) IP Spoofing 4) Broadcasts Request	Properly configure /etc/sysctl.conf

Table 3: Server Security vulnerabilities and countermeasures.

Sr No	Vulnerability	Attacks	Countermeasure
1.	**FTP** i) Anonymous access ii) Too many user access	1) Unauthorized access 2) Denial of Service Attacks (DoS*)*	1) Apply proper security parameter 2) Apply DOS security parameters
2.	ssh password	Cracking of password	Use passphrase
3.	Unauthorized websites	Unauthorized access	Authenticate the website

IV. PROPOSED SYSTEM

The Linux security is centered on how the configuration is made. Configuration files for various system processes, application and servers play the vital role in hardening the Linux. Configuration file contains various security related attributes that need to be considered while at the time of configuration of particular application, process and server.

The authors have focused on various configuration files that are critical from security perspective and security attributes present in such configuration files. The following Fig. 1.1 shows the detail description about how to make Linux more secure so that impact of security breach can be minimum.
The Linux Hardening model consists of three modules which makes the Linux more secure from the attackers which are:

1. Vulnerability check module
2. Log Analysis Module
3. Security Module

1. Vulnerability check module

As Mentioned in the literature survey there are various configuration files such as system configuration file and server configuration files which contains attributes that are critical. This module will check such configuration files and scan for attribute which are important from security perspective. This module check current attribute value with best security value required for that attribute. If current configured value is not a best security value then it will consider it as vulnerability and generates the vulnerability report. Generated report is given to the security module.

2. Log Analysis Module

Linux system consists of very strong logging mechanism maintains the log for kernel, servers, users, system processes etc. These entire logs by default placed at different location. This module collects the log from these various places and generates report. This generated report is useful for finding the vulnerability. Generated report is given to the security module.

3. Security Module

This module collects the vulnerability report and log analysis report and applies security. By looking vulnerability report this module get the vulnerable configuration files and modify them with best security practice. Similarly by looking log analysis report this module apply the security attributes accordingly. This model is actually responsible for modifying the configuration files and making the Linux more secure.

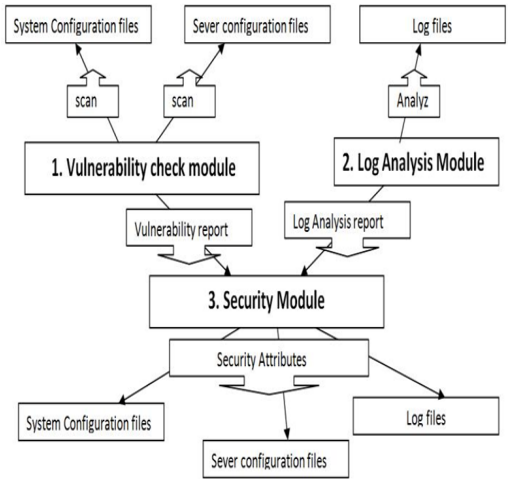

Figure : Linux Hardening Model

V. CONCLUSION

To increase reliance on powerful, networked computers to help run businesses and keep track of personal information, entire industries have been formed around the practice of network and computer security. Enterprises have solicited the knowledge and skills of security experts to properly audit systems and tailor solutions to fit the operating requirements of their organization Most of the organizations are increasingly dynamic in nature, their workers are accessing critical company IT resources locally and remotely,

hence the need for secure computing environments has become more pronounced. This paper describes how to simply, consistently, and practically secure the Linux environment.

Thus the Linux security is centered on proper system.

REFERENCES:

[1] P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner, and J. F. Farrell ," The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments ",21st National Information Systems Security Conference, NSA, 1998,PP 303– 314

[2] C. J. PeBenito, F. Mayer, and K. MacMillan. Refer-ence Policy for Security Enhanced Linux. In SELinux Symposium, 2006.

[3] R. Wita and Y. Teng-Amnuay. Vulnerability profile for linux. In Proceedings of the 19th International Conference on Advanced Information Networking and Applications, pages 953–958. IEEE, 2005.

[4] R. Spencer, S. Smalley, P. Loscocco, M. Hibler, D. An-dersen, and J. Lepreau. The Flask Security Architec-ture:System Support for Diverse Security Policies. In The Eighth USENIX Security Symposium, pages 123–139, August 1999.

[5] Nigel Edwards, Joubert Berger, and TseHoungChoo. A Secure Linux Platform. In Proceedings of the 5th Annual Linux Showcase and Conference, November 2001

[6] Crispin Cowan, Steve Beattie, Calton Pu, PerryWagle, and Virgil Gligor. SubDomain: Parsimonious ServerSecurity. In USENIX 14th Systems Administration Conference (LISA), New Orleans, LA, December 2000.

[7] Red hat enterprise linux 6 security guide (Red Hat
Engineering Content Services).

[8] Afinidad, T. E. Levin, C. E. Irvine, and T. D.Nguyen, "A model for temporal interval authorizations," inHawaii International Conference on System Sciences, Software Technology Track, Information Security Education and Foundational Research, (Kauai, Hawaii), p. to appear, January 2006.