Linux Hardening

Namita Arora
Dept of comp science
PES's Modern College of Engineering
Pune, India
arora.namita5@gmail.com

Tejasvi Bhosale
Dept of comp science
PES's Modern College of Engineering
Pune, India
tejasvi.bhosale944@gmail.com

Vishakha Sharma
Dept of comp science
PES's Modern College of Engineering
Pune, India
vishakha.sharma@gmail.com

Jyoti Supe
Dept of comp science
PES's Modern College of Engineering
Pune, India
supejyoti86@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 paper also presents onto detect the vulnerabilities in the system by scanning configuration files and server files, to determine the computer activities by scanning the log files thereby securing the system by replacing the vulnerable attributes with secured attributes. Application security is ensured by scrutinizing the signatures of various applications and displaying all the functionalities in GUI format making it more user friendly. A very important step in securing a Linux system is to determine the primary function or role of the Linux server. You should have a detailed knowledge of what is on your system. Otherwise you will have a difficult time to understand what needs to be secured and hence securing your Linux system proactively won't be that effective. Therefore it is very critical to look at the default list of software packages that don't comply with your security policy. If you do that you will have less packages to update and to maintain when security alerts and patches are released.

Keywords*-vulnerability; package; netfilter; TCP wrapper; hardening.*

*****

I. INTRODUCTION

The term "Hardening" refers to securing the system. Like any other operating system, application level security flaws leave Linux vulnerable to a variety of malicious attacks. Over the years, many tools and techniques have been developed to "harden" Linux hosts in an attempt to mitigate the risk posed by buggy software. The Linux operating system has numerous settings and permissions which provide high degree of customization but this same feature also makes it a challenge to secure properly. Add to this nature of the open source software environment where anybody can create a program for this operating system and you end up with an infinite number of possible configurations. The goal is to use these properties to create a secure system that meets the user's needs.

II. IMPLEMENTATION

In order to ensure complete security, the Linux operating system must be secured from the following aspects:

- User Security
- Network Security
- Package Security

A. User Security

1) Vulnerability assessment: A vulnerability assessment is an internal audit of your network and system security; the results of which indicate the confidentiality, integrity, availability of your network. Typically, vulnerability assessment starts with a reconnaissance phase, during which important data regarding the target systems and resources are gathered. This phase leads to the system readiness phase, whereby the target is essentially checked for all known vulnerabilities. The readiness phase culminates in the reporting phase, where the findings are classified into categories of high, medium, and low risk; and methods for improving vulnerability. The security (or mitigating the risk of vulnerability) of the target are discussed.

The following are some of the benefits of performing vulnerability assessments:

- Creates proactive focus on information security.
- Finds potential exploits before crackers find them.
- Results in system being kept up-to-date and patched.
- Promotes growth and aids in developing staff expertise.
- Abates financial loss and negative publicity.

TABLE 1 VULNERABILITIES

Vulnerability	Attacks	Countermeasure
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
Unnecessary software's	Software vulnerability attack	Install minimum software's
maliciously altered package	System instability ,System crash and data loss, data still	Install Signed Packages
No BIOS password	Stealing/Changing Data Using a Bootable Linux CD	Give BIOS password
Single User Mode access	Access as root user without password	Password protecting BIOS
Access to the GRUB Console	change its configuration or to gather information using the ***cat*** command.	Password protecting GRUB
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
Weak password, no password or default password	Cracking of weak passwords	1) Enforcing Stronger Passwords 2) Restricting Use of Previous Passwords 3) Locking User Accounts After Too Many Login Failures
No password Aging	Use of Cracked password over long period of time	Apply good password Aging
root access to individual users	1)Machine Misconfiguration 2)Running Insecure Services	1) Root Disallowing Access 2) Disallow Remote Root Login 3) Disabling root access via any console device (tty)
Allowed *su* command to users	Access other user data and services	Limit and block *su* access
Enabled CTRL-ALT-Delete	Unauthorized System Shut down	Disable CTRL-ALT-Delete
OS fingerprinting	Get os information like OS version etc.	Place login banner
Local log monitoring	Remove of log entries and log files	Remote log monitoring
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

2) Password security: Passwords are the primary method that red hat enterprise Linux uses to verify the users identity. This is why password security is so *important for the protection of the user, the workstation and the network. Password aging is one technique used by system administrators to defend against bad passwords within an organisation. password aging means that after a specified period (usually 90 days), the user is prompted to create a new password. The theory behind this is that if a user is forced to change his password periodically, a cracked password is only useful to intruder for a limited amount of time.*

3) Log monitoring: In order to determine ongoing operational status of your system and applications, log monitoring plays an important role. When it comes to security, we need to delve a bit deeper into the logging world to gain a clearer understanding of what is going on with our system and applications and thus identify potential threats and attacks. Logs are also key targets for someone who wants to penetrate your system-for two reasons:

- The first reason is that your logs often contain vital clues about the system and its security. Attackers often target the logs in an attempt to discover more about the system. As a result, we need to ensure our log files and /var/log directory are secure from intruders and that log files are available only to authorized users. Additionally, if you transmit your logs over the network to a centralized log server, you need to ensure no one can intercept or divert your logs.
- The second reason is that if attackers do penetrate your systems, the last thing they want to happen is that you detect them and shut them out of your system. One of the easiest ways to prevent you from seeing their activities is to whitewash your logs so that you see only what you expect to see. Early detection of intrusion using log monitoring and analysis allows you to spot them before they blind you.

B. Network security

Potentially, any network service is insecure. This is why turning off unused services is so important. Some network protocols are inherently more insecure than others. These include any services that: Transmit usernames and Passwords Over a Network Unencrypted –Many older protocols, such as Telnet and FTP, do not encrypt the authentication session and should be avoided whenever possible. Firewall is an important measure to protect network security. Firewall can be used to enhance access control between two or more networks. The Linux kernel

 ISSN: 2321-8169
1019 – 1022

uses the netfilter facility to filter packets, allowing some of them to be received by or pass through the system while stopping others. This facility is built in to the Linux kernel, and has three built in tables or rules lists, as follows:

- Filter- the default table for handling network packets.
- Nat-used to alter packet that create a new connection and used for Network Address Translation (NAT).
- Mangle- used for specific types of packet alteration.

The built in chains for the filter table are as follows:

- Input- applies to network packets that are targeted for the host.
- Output- applies to locally generated network packets.
- Forward- applies to network packets routed through the host.

For network services that utilize nefilter, TCP Wrappers add an additional layer of protection by defining which hosts are or are not allowed to connect to "wrapped" network services. One such wrapped network service is the xinetd super server. This service is called a super server because it controls connections to a subset of network services and further refines access control.

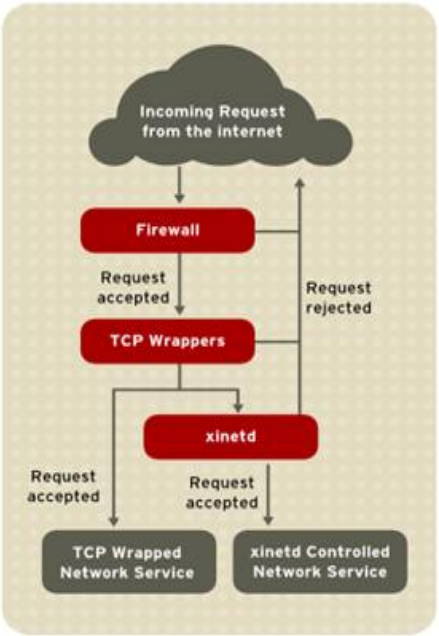

Fig.1 Access Control to networked services

The TCP Wrappers (tcp_wrappers and tcp_wrappers-libs) are installed by default and provide host-based access control to control to network services. When a connection attempt is made to a TCP-wrapped service, the service first references the host's access files (/etc/hosts.allow and /etc/hosts.deny) to determine whether or not the client is allowed to connect. Because TCP Wrapper are a valuable addition to any server administrator's arsenal of security tools, most network services within Red Hat Enterprise Linux are linked to the libwrap.solibrary. Such application include /usr/sbin/sshd, /usr/sbin/sendmail, and /usr/sbin/xinetd. To determine if a client is allowed to connect to a service, TCP Wrappers reference the following two files which are commonly referred to as hosts access files:

- /etc/hosts.allow
- /etc/hosts.deny

When a TCP-wrapped service receives a client request, it performs the following two steps:

1. It references etc/hosts.allow – The TCP-wrapped service sequentially parses the /etc/hosts.allow file and applies the first rule specified for that service. If it finds a matching rule, it allows the connection . if not, it moves on to the next step.
2. It references /etc/hosts.deny – the TCP-wrapped service sequentially parses the /etc/hosts.deny file. If it finds a matching rule, it denies the connections. If not, it grants access to the service.

The xinetd daemon is a TCP-wrapped super service which control access to a subset of popular network services, including FTP,IMAP and Telnet. It also provides service-specific configuration options for access control, enhanced logging, binding, redirection and resources utilization control. When a client attempts to connect to a network service controlled by xinetd, the super service receives the request and checks for any TCP Wrappers access control rules. If access is allowed , xinetd verifies that the connection is allowed under its own access rules for that service. It also checks that the service is able to have more resources assigned to it and that it is not in breach of any defined rules.

C. Package security

RPM is an open packaging system, which runs on Red Hat Enterprise Linux as well as on Linux & UNIX system. The utility works only with packages built for processing by the rpm package. RPM maintains a database of installed packages & their files. This software provides the functionality to retrieve the number of packages installed and their verification reports are generated. Software packages are published through repositories. All well-known repositories support package signing. Package signing uses public key technology to probe that the package that was published by the repository has not been changed since the signature was applied. This provides some protection against installing software that may have been maliciously altered

after the package was created but before you downloaded it. Using too many repositories, untrustworthy repositories, or repositories with unsigned packages has a higher risk of introducing malicious or vulnerable code into your system. It is very critical to look at the default list of software packages and remove unneeded packages or packages that don't comply with your security policy. It is best practise to install only the packages you will use because each piece of software on your computer could possibly vulnerability.

III. CONCLUSION

Our main contribution is in designing and building a secure file system and network that was developed with the express goal of enhancing file data security and network security in Linux kernel. The main objective is to detect the vulnerabilities in the system by scanning configuration file and server files , to determine the computer activities by scanning the log files thereby securing the system by replacing the vulnerable attributes with secured attributes. In network security we provide security for web server ssh server etc. application security is ensured by scrutinizing the signature of various applications and displaying all the functionalities in GUI format making it more user friendly.

REFERENCES

[1] Deng Yiquan, "Linux Network Security Technology", IEEE computer, 978-14577-0860-2/11,2011

[2] Udi Ben-Porat, Student Member,IEEE,Anat Bremler-Barr,Member,IEEE,and Hanoch levy,Member,IEEE,"Vulnerability of Network Mechanisms to Sophisticated DDoS Attacks"0018-9340/13,IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 5, MAY 2013.

[3] Terry Collings & Kurt Wall, "Red_Hat_Enterprise_Linux-6-Deployment_Guide-en-US"2002 access.redhat.com

[4] McGraw-Hill Companies, "hacking exposed Linux: Linux security secrets & solutions", 2008 www.barnesandnoble.com

[5] Red Hat Engineering Content Services, "Red_Hat_Enterprise_Linux-6-Security_Guide-en-US",2011 access.redhat.com