Linux Security

Daniel Graves, Francis Ortiz, Harrison Brown, Larry Altman
Saint Leo University

Abstract

In our paper we are talking about Linux Security and all of the overall functions of it and how it works. There are many different sections that each have a key part where it talks about the type of security that is being used and how it is used overall in the system. We have the introduction where we provide a brief history of Linux security. We then have five main topics that cover all the different areas in Linux security. First, we have the **importance of security** in general and how it is changing drastically throughout all these different companies and around the world. Second, we have **discretionary access control** that meet certain requirements for Linux but changes drastically all the time. Third we have **namespaces** which is great for implementing security and is used for other sources of security. Fourth we have **network security** where we talk about how important it is to Linux security and the interfaces. Last, we discuss **cryptography** where is everything from hashing to IPsec. The Linux kernel is very useful for all of these main reasons and is used widely today throughout may companies and in different type of software.

Linux is a phenomenon of the Internet. Born out of the hobby project of a student it has grown to become more popular than any other freely available operating system. Universities and research establishments use it for their everyday computing needs. It is a fully developed and professionally written operating system used by enthusiasts all over the world.

The Linux kernel, created by Linus Torvalds and released in 1991, was made available to the world for free. Torvalds then invited others to add to the kernel provided they keep their contributions free. Thousands of programmers began working to enhance Linux, and the operating system grew rapidly. Because it is free and runs on PC platforms, it gained a sizeable audience among hard-core developers quickly. Linux has a dedicated following and appeals to several types of users.

The Importance of Security

Our computers have become an extension of everything we do from banking and investing to shopping and communicating with others through email or chat. You may not consider your communications "top secret," but you may not want strangers reading your email, using your computer to attack other systems, sending forged email from your computer, or examining personal information stored on your computer.

Computer security is the process of preventing and detecting unauthorized use of your computer. Prevention measures help you stop unauthorized users (hackers) from accessing any part of your computer system. Detection helps you to determine whether someone attempted to break into your system, if they were successful, and what they may have done.

Hackers do not care about your identity. Often, they want to gain control of your computer, so they can use it to launch attacks on other computer systems. Having control of your computer gives the hackers the ability to hide their actual location as they launch attacks, often against high-profile computer systems such as government or financial systems. Hackers can watch all your actions on the computer, or cause damage to your computer by reformatting your hard drive or changing your data. Unfortunately, hackers are always discovering new vulnerabilities to exploit in computer software. The complexity of software makes it increasingly difficult to thoroughly test the security of computer systems.

Security should be one of the foremost thoughts at all stages of setting up your Linux computer. To implement a good security policy on a machine requires a good knowledge of the fundamentals of Linux as well as some of the applications and protocols that are used. Although Linux users are less prone to viruses than some other major operating systems, there are still many security issues facing Linux users and administrators.

Discretionary Access Control

Linux was initially developed as a clone of the Unix operating system in the early 1990s. As such, it inherits the core Unix security model—a form of Discretionary Access Control (DAC). The security features of the Linux kernel have evolved significantly to meet modern requirements, although Unix DAC remains as the core model.

Unix DAC allows the owner of an object (such as a file) to set the security policy for that object, which is why it's called a discretionary scheme. As a user you can create a new file in your home directory and decide who else may read or write to the

file. This policy is implemented as permission bits attached to the file's inode, which may be set by the owner of the file. Permissions for accessing the file, such as read and write, may be set separately for the owner, a specific group, and other (i.e. everyone else). This is a relatively simple form of access control lists (ACLs).

Programs launched by a user run with all the rights of that user, whether they need them or not. There is also a superuser, an all-powerful entity which bypasses Unix DAC policy for managing the system. Running a program as the superuser provides that program with all rights on the system.

Unix DAC is a relatively simple security scheme, although, designed in 1969, it does not meet all the needs of security in the Internet age. It does not adequately protect against buggy or misconfigured software, for example, which may be exploited by an attacker seeking unauthorized access to resources. Privileged applications, those running as the superuser (by design or otherwise), are particularly risky in this respect. Once compromised, they can provide full system access to an attacker.

Functional requirements for security have evolved over time. Extensions have been developed to enhance protection in Linux systems without compromising existing functionality. The ideal solution would be to build a new security system from the ground up, but it is not viable since the entire operating system would also have to be rebuilt to work with this new security system.

Namespaces

Namespaces in Linux derive from the Plan 9 operating system (the successor research project to Unix). It's a lightweight form of partitioning resources as seen by processes, so that they may, for example, have their own view of filesystem mounts or even the process table. This is not primarily a security feature but is useful for implementing security. One example is where each process can be launched with its own, private /tmp directory, invisible to other processes, and which works seamlessly with existing application code, to eliminate an entire class of security threats.

The potential security applications are diverse. Linux Namespaces have been used to help implement multi-level security, where files are labeled with security classifications, and potentially entirely hidden from users without an appropriate security clearance.

Network Security

Linux has a comprehensive networking stack, supporting many protocols and features. Linux can be used both as an endpoint node on a network, and as a router, passing traffic between interfaces according to networking policies.

Netfilter is an IP network layer framework which hooks packets which pass into, through and from the system. Kernel-level modules may hook into this framework to examine packets and make security decisions about them. *iptables* is one such module, which implements an IPv4 firewalling scheme, managed via the userland iptables tool. Access control rules for IPv4 packets are installed into the kernel, and each packet must pass these rules to proceed through the networking stack. Also implemented in this codebase is stateful packet inspection and Network Access Translation (NAT). Firewalling is similarly implemented for IPv6.

ebtables provides filtering at the link layer and is used to implement access control for Linux bridges, while *arptables* provides filtering of ARP packets. The networking stack also includes an implementation of *IPsec*, which provides confidentiality, authenticity, and integrity protection of IP networking. It can be used to implement VPNs and point to point security.

Cryptography

A cryptographic API is provided for use by kernel subsystems. It provides support for a wide range of cryptographic algorithms and operating modes, including commonly deployed ciphers, hash functions, and limited support for asymmetric cryptography. There are synchronous and asynchronous interfaces, the latter being useful for supporting cryptographic hardware, which offloads processing from general CPUs.

Support for hardware-based cryptographic features is growing, and several algorithms have optimized assembler implementations on common architectures. A key management subsystem is provided for managing cryptographic keys within the kernel. Kernel users of the cryptographic API include the IPsec code, disk encryption schemes including ecryptfs and dm-crypt, and kernel module signature verification.

These are only some of the tools and features that Linux uses to protect data integrity. The kernel security has evolved from its Unix roots, adapting to ever-changing security requirements. These requirements have been driven both by external changes, such as the continued growth of the Internet and the increasing value of information stored online, as well as the increasing scope of the Linux user base. Ensuring that the

security features of the Linux kernel continue to meet such a wide variety of requirements in a changing landscape is an ongoing and challenging process.

References

https://computer.howstuffworks.com/question246.htm

https://www.thenetworkpro.net/2010/04/02/why-is-computer-security-important/

https://www.linux.com/learn/overview-linux-kernel-security-features

https://www.tecmint.com/linux-server-hardening-security-tips/