Linux Authentication and PAM

Face Recognition

Henry Roeth

2024-02-16

Abstract

In computer security, authentication is the process of confirming someone is who they claim to be when attempting to access any kind of computer system through the confirmation of something they have, something they know, or something they are. It is important to always maintain and improve the integrity of computer systems' authentication schemes as new security threats arise. The Linux kernel invokes the standard Unix authentication process across the majority of its applications. However, as new forms of authentication are developed, it is inefficient to individually reconfigure applications such that the desired authentication scheme is incorporated. The PAM (Pluggable Authentication Module) mechanism in Linux integrates various low-level authentication schemes into a high-level API, allowing programs that require some form of authentication to be developed independently from the desired authentication scheme. This integrated research aims to demonstrate the function and importance of PAM in Linux authentication with the invocation of a face recognition PAM security module. Real-time face detection and recognition is performed using the Haar Cascade Classifier and the LBPH (Local Binary Patterns Histograms) feature-based face recognition method.

Introduction

The Viola and Jones (2001) Linux Ahonen et al. (2004) kernel Geisshirt (2007) invokes the standard Unix authentication process across most of its applications (e.g., common-auth, sshd, su, and sudo). When developers are creating and updating applications, it would prove quite inefficient to individually reconfigure application logic such that it aligns with the newly desired authentication scheme. This would mean restructuring code and requiring other dependencies. With this in mind, developers need a way to invoke authentication schemes independently from applications, allowing for a more modular approach. This enables programs to run separately. Much like it's father kernel (Unix) the Linux kernel invokes these modular authentication schemes via the PAM (Pluggable Authentication Module) mechanism. One might think of this as a multitool. Just as a multitool can have different tools for various purposes like cutting, screwing, or opening, PAM provides different authentication methods for Linux. Depending on what you need to authenticate—be it through passwords, biometrics, or other means—you can plug in the appropriate tool/module into PAM to handle the authentication process effectively. This integrated research aims to display this function with the creation of a modular authentication scheme—face recognition.

Overview of Linux Authentication

Linux authentication involves several components. Centrally are the /etc/passwd and /etc/shadow files. In /etc/passwd, user account information such as usernames, user IDs, group IDs, and home directories are stored. The tangible passwords are more securely stored in /etc/shadow. It is here where there are hashed passwords and other security-related information. Each line in /etc/shadow represents a user account and includes uniquely populated fields like usernames, encrypted passwords, password aging and expiration details, and account lockout information. When any user attempts to login to the machine, the system hashes the entered password using the same algorithm as the one stored in /etc/shadow. If the hashed passwords match, access is granted to the user. More specifically, hashing involves the conversion of a password into a fixed-length string of characters using a cryptographic hash function. This process is irreversible, meaning the original password cannot be easily derived from the hash. Another additional measure of security that is implemented in the hashing process is salting. Salting is where a random value (a salt) is added to the password before hashing. This prevents attackers from using precomputed hash tables, also known as rainbow tables, to crack passwords. These unique fields in the /etc/shadow file are separated by colons, with each field serving a specific purpose. These fields may typically include the username, the hashed password, the last password date, the minimum and maximum password age, the password warning period, the password inactivity period, the account expiration date, and the account expiration date warning. To summarize, the integrity of the Linux authentication process relies on secure data storage of hashed passwords, salting for additional security, and an automated management system of user account information. As this relates to PAM, the invoked module (found in the directory /lib/*/security) is the primary logic to which the input password is hashed and compared to the corresponding hash in the /etc/shadow file; it is then followed by either a success or failure being return to the PAM mechanism.

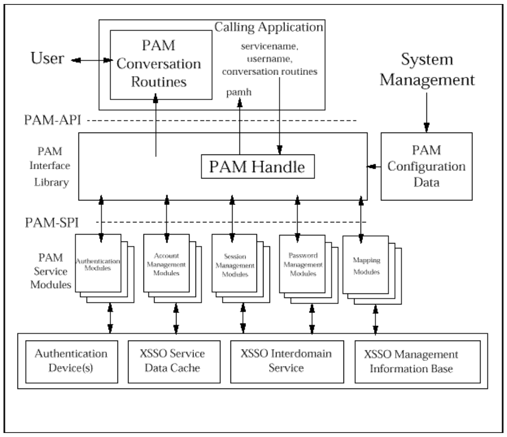

Figure 1: Example image 1 with a caption.

PAM Configuration

The PAM mechanism generally handles all authentication in Linux.

Authentication and Biometrics

To be continued.

References

Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns.

Geisshirt, K. (2007). *Pluggable Authentication Modules: The Definitive Guide to PAM for Linux Sysadmins and C Developers*. Packt Publishing, Birmingham, UK.

Viola, P. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511.