INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 144-149

e-ISSN : 2583-1062

Impact Factor : 5.725

REMOTE ACCESS PROTECTION FOR LINUX MACHINES

Dr. S. Satheesbabu[*1], Akinthiya Srinath K I[*2], Arunsnalan R[*3], Baranie M[*4], Hariharan B[*5]

[*1]Professor, Department of CSE, PSNA CET, Dindigul, TamilNadu, India.

[*2,3,4,5]Students, Department of CSE, PSNA CET, Dindigul, TamilNadu, India.

ABSTRACT

Majority of enterprise machines which handle confidential data are Linux systems. These Linux systems are made to be publicly accessible by SSH or putty to customers, clients, and developers. Current Linux authentication only includes just username and password which is considered vulnerable today for many attacks like phishing, shoulder surfing etc. Enforcement of multifactor authentication by directly hooking the Linux authentication can help enhance security multi fold as added authorization layer helps identify personalities. Special logging techniques and custom IP level security can also be enforced by the proposed hooking method. This methodology of hooking Linux authentication can not only be used for Linux logins but also services that use Linux authentication which includes SSH, sudo, Identity providers like Active Directory.

Keywords: PAM, authentication, MFA, Linux machine.

1. INTRODUCTION

Every Linux machine irrespective of its distro has the authentication module which acts as the entry point in perspective of an user whether be the owner or a remote user. As the authentication module employed in popular distros is just a plug in that collects credentials from the user that are necessary and runs it against the data that is stored already in the registry, Neglecting the individual software and service authentications, this entry level authentication is enough for anyone to access an up and running Linux machine. So, in summary the only requirement for any kind of user to access a Linux machine is a pair of legitimate username and password. This information is in the hands of a user. There is no added layer of protection. It is so easy for this sensitive piece of information to leak out if the user is vulnerable. Even when ruling out the vulnerabilities, the credentials might linger with a remote user when it was supposed to be a onetime access for the Linux machine. From the above description it becomes evident that there is a lingering need for an optional but additional layer of protection. It could be a simple authorization. But before establishing the authorization part, there must be a proper depiction on how the existing authentication module works and what is flow of data during the authentication process. Establishment of this concept will come in handy during the authorization flow.

2. PROBLEM STATEMENT

In recent years, cloud computing has emerged as an efficient way for individuals and organizations to store, process, and access data on the internet. Public cloud environment offers several advantages such as scalability, cost effectiveness, and accessibility. However, the shared responsibility model of cloud security means that the cloud service provider is only responsible for the security of the cloud infrastructure, leaving the responsibility of securing the data on the customer and particularly in public clouds where data is shared with multiple users. With the increasing reliance on public cloud services, it is becoming the critical to develop effective solutions for secure data storage on public clouds. Therefore, the challenge is to develop a model for secure data storage on public cloud that ensures the confidentiality, integrity, and availability of data, while also complying with regulatory and compliance requirements. The model must be able to protect data from unauthorized access, data breaches, and loss while maintaining the usability and accessibility of the data. It should consider the performance and cost implications of implementing security measures on the cloud infrastructure.

3. EXISTING SYSTEM

Google Authenticator is a popular MFA tool that generates time-based one-time passwords (TOTP) on your smartphone. It can be used to secure SSH logins, sudo access, and other services on Linux systems. The setting up of google authenticator as the MFA provider of Linux involves downloading their pam package lib pam-google-authenticator and configuring the pam Duo Security is a cloud-based MFA service that provides a range of authentication methods, including push notifications, SMS, phone calls, and hardware tokens. It can be integrated with SSH, PAM, and other Linux services. To use Duo Security on Linux, you need to sign up for a Duo account, install the Duo Authentication Proxy, and configure it to work with your Linux server. Azure AD Multi-Factor Authentication is a cloud-based MFA service that provides a range of authentication methods, including phone calls,

text messages, mobile app notifications, and hardware tokens. It can be integrated with Linux servers using the PAM module. To use Azure AD Multi-Factor Authentication on Linux, you need to sign up for an Azure AD account, configure the MFA settings for your users, and install the PAM module on your Linux server YubiKey is a hardware-based MFA device that provides a range of authentication methods, including one-time passwords, smart card authentication, and FIDO2/Webathons. It can be used to secure SSH logins, sudo access, and other services on Linux systems. To use YubiKey on Linux, you need to install the YubiKey-manager package and configure it to work with your SSH server.

4. DRAWBACKS OF EXISTING SYSTEM

- **Password-based authentication:** Passwords are the most common form of authentication in Linux, but they have several drawbacks. Passwords can be weak and easy to guess, and they can also be forgotten or lost.
- **Lack of two-factor authentication:** Many Linux authentication systems do not support two-factor authentication, which adds an additional layer of security by requiring users to provide two forms of identification.
- **Single point of failure**: Some authentication systems have a single point of failure, such as a central authentication server, which can be vulnerable to attack.
- **Complexity**: Some authentication systems can be complex to set up and manage, especially if they involve multiple servers or services.
- **Lack of scalability**: Some authentication systems may not scale well, especially in large organizations where there may be thousands of users who need to be authenticated.
- **Limited Support**: Some authentication systems may have limited support or may not be well maintained, which may lead to security bugs or compatibility issues with other software.

5. PROPOSED SYSTEM

The proposed system focuses mainly on the easy installation and high customization of the MFA according to the user needs. The proposed system is a multilayer architecture each simplifying the process of installation at the same time in the motive to bring high customization ability. This system involves an API which connects to database which has the user customization settings. The installation script module will take care of all the necessary installation changes so that the user need not have much fatigue while installation. Instead of complicated installation which follows in many popular apps like google authenticator and DUO the proposed system provides a one clicks installation and configuration approach.

- The proposed system is focused on providing high compatibility over customization of MFA which when needed on the previous methodologies will take lot of manual user steps which is totally avoided
- Also, the proposed solution helps in easy integration to existing MFA providers like google and duo with same less or no configuration needed.

The implementation utilizes the property of PAM (Pluggable Authentication Module) to provide a customizable multifactor authentication security feature to the Linux users without much intervention by the users. The process of authentication involves the following steps included from user to the system.

- User registers for the MFA they need
- Download the Linux client script
- Client script installs the client program and pam modules
- The client script automatically configures the necessary pam files adding the pam modules depending upon the user requirements on MFA
- When the user logins on the next time the a pre-auth check happens to verify the user
- The API gets the call and does the pre-auth and sends the validation response which contains the chosen MFA.
- Depending upon the response the client script the client script triggers the PAM file to prompt the user for MFA answer (i.e., TOTP, Security Question, verification code etc.…)
- The entered result will be sent back to API for verification and once the client script gets the result as verified it sends the notification to PAM file which in turn validates the user by sending the PAM_SUCCESS flag.
- Thus, the user is authenticated through MFA

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor : 5.725

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 144-149

6. FEASIBILITY STUDY

Every system is feasible if they have unlimited resources and infinite time. The primary objective of the feasibility study remains testing the Technical, Operational and Economical feasibility for adding new modules. All systems are feasible if they have unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

6.1 TECHNICAL FEASIBILITY

Frequently asked questions during a feasibility study include:

- Are there appropriate tools for implementing the recommendations?
- Is the equipment under preparation capable of storing the information needed to implement the new system
- Will the app adequately answer questions regardless of the number or location of users?
- Is it possible to change the system after creating it?
- Is there assurance of accuracy, reliability, availability and data security?

Early systems did not exist to meet "security infrastructure" requirements.

6.2 OPERATIONAL FEASIBILITY

A project plan is only useful if it can be turned into information. This will be done according to the work of the institution. The effective implementation of the project will be considered as an important part of the implementation of the project. Some important questions to ask to evaluate the effectiveness of the program are:

Are there enough users to support the administration?
If it is built and used, will it be used and will it work properly?
Will there be objections from users that will affect the results of the application? The system is designed for the above problems. Administrative and user problems should be considered first.

So there is no doubt that user protection will destroy a good app. A well-planned program will ensure efficient use of computer resources and help increase performance.

6.3 ECONOMICAL FEASIBILITY

It is technically possible to create a system that will be a good investment for the organization if it is installed and used. In terms of economic potential, the development costs of building a system are judged by the best results of the new system. The financial benefits must equal or exceed the costs. The system is economically viable. It does not require any additional hardware or software.

This system has some economic and financial benefits as it is interfaced using existing resources and technology that t he NIC can use.

7. METHODOLOGY

7.1 Installation Script

The installation scripts install the necessary dependency packages and the client script which acts as the communication module between the server API and the client Linux machine. The installation script also installs the PAM module and configure the application specific PAM files to push the MFA on to the application making it mandatory for the users to go through the MFA tunnel to get Authenticated. The installation script also verifies the user once again during installation for additional security to API calls. Which is a onetime verification.

7.1 Client Script

The client script is the communication module which communicates between the PAM module and the server API to send and receive request and response. The main activity of the client script is to coordinate between the pam module which interacts with the user and authentication and server which holds the user and MFA data. The client starts with the pre-authentication of the user by sending the user details to the server API and initiates the MFA procedure by intimating the PAM module. The script gets the user entered security answer and sends it to the server API for verification. And once verified it intimates the PAM module which then authorizes the user.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 144-149

e-ISSN : 2583-1062

Impact Factor : 5.725

7.2 PAM Module

The pam module is the authentication module which when the set condition is correct will authenticate the user and deny login if not. The advantage of using the pam module is the customization feature provided by PAM. Each linux authenticated application like SSH, sudo uses PAM specific file which loads all the necessary modules for the authentication. The PAM module which is installed by the installation script will also be added to the user desired application pam file.Once added, the pam module will act as the communication between user authentication module and the client script. The pam module uses PAM CONVERSE to prompt user and receive the security answer which then will be sent to the client script for verification. The pam module approves/rejects the authentication by returning flags.

- PAM_SUCCESS flag signifies the success of condition
- PAM_FAILURE flag signifies the failure of condition
- PAM_ABORT flag signifies to abort the program without any retries

Benefits of using this solution include increased security by adding an additional layer of authorization, enhanced protection against attacks such as phishing and slashing, and the ability to identify itself. Special decision-making techniques and dedicated IP\level security also helps detect and prevent attacks.
The solution is also flexible as it can be used for many services that use Linux authentication.

8. SYSTEM ARCHITECTURE

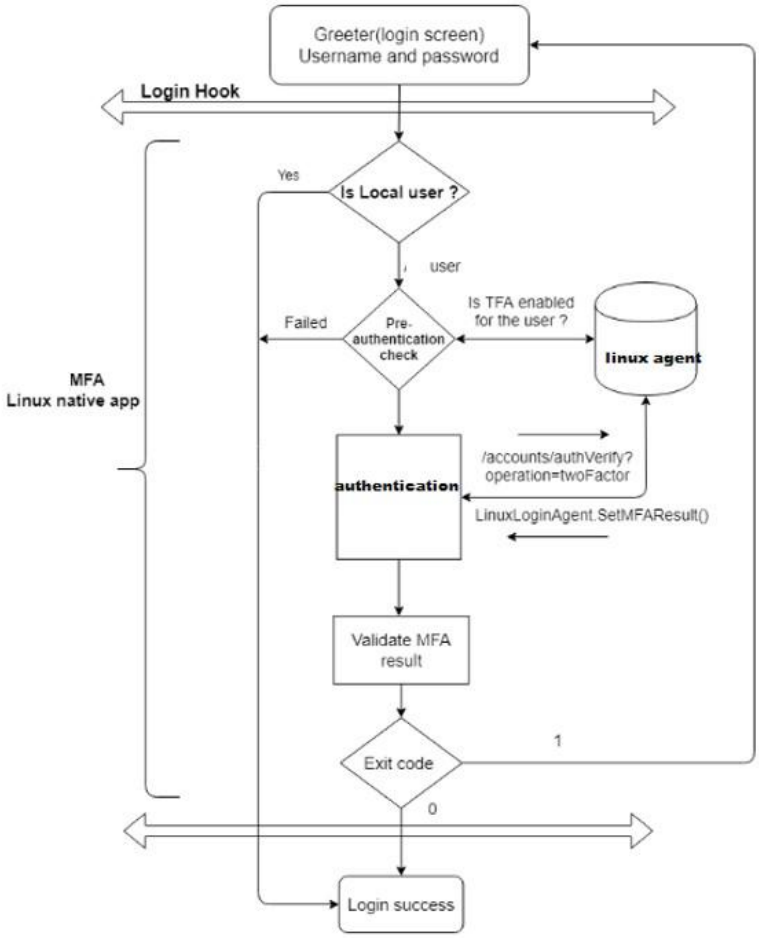

Figure 1: System architecture diagram

9. RESULTS AND DISCUSSION

9.1 Verification by TOTP

TOTP is a simple and effective way to add an extra layer of security to your online accounts. It provides an additional barrier to unauthorized access, reducing the risk of data breaches and identity theft. However, it is important to note that no security system is foolproof, and TOTP should be used in conjunction with other security measures such as strong passwords and regular security updates.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 03, Issue 04, April 2023, pp : 144-149

e-ISSN : 2583-1062

Impact Factor : 5.725

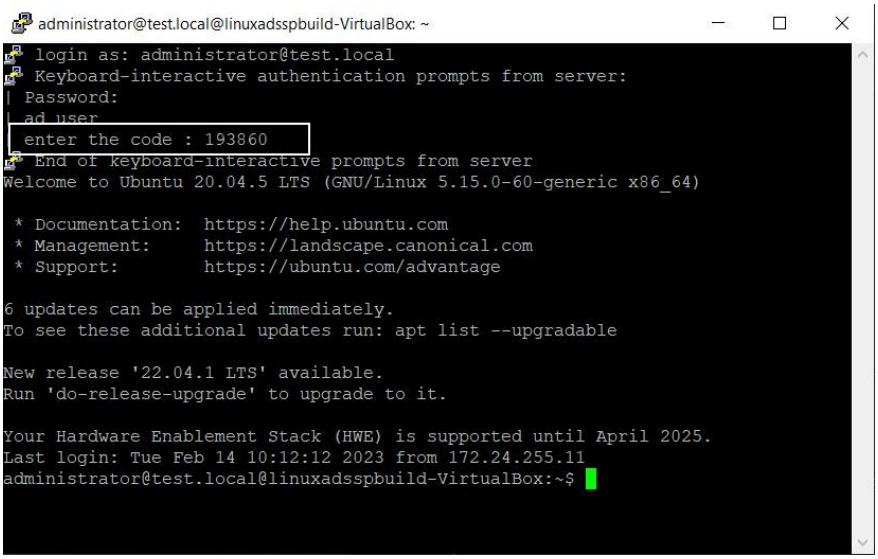

Figure 2: TOTP Verification at user endpoint

9.1 Downloading Installation File

The installation scripts install the necessary dependency packages and the client script which acts as the communication module between the server API and the client Linux machine. The installation script also installs the PAM module and configure the application specific PAM files to push the MFA on to the application making it mandatory for the users to go through the MFA tunnel to get Authenticated. User has to Register and select the required MFA and click on Get File to get an Installation Scripts

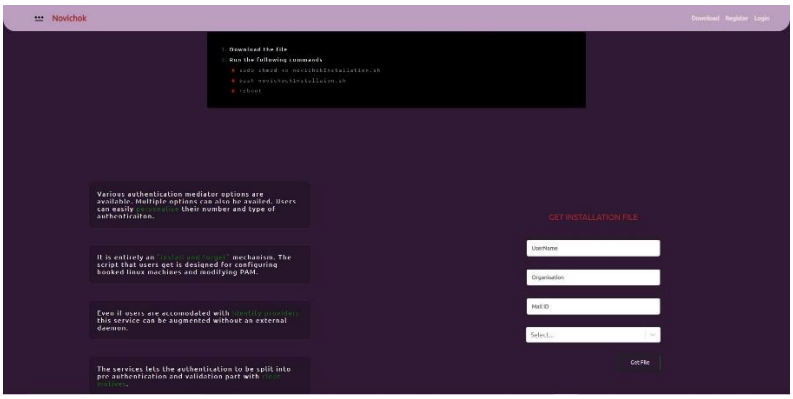

Figure 3: Web app

10. CONCLUSION

As a result, the current Linux authentication system can be used to prevent phishing, shoulder snooping, etc. It is vulnerable to various attacks such as This is where an extra layer of protection is needed. The request link to the Linux authentication method can be used to manage multiple authentications and provide an additional layer of authentication. This hooking method can also be used for services that use Linux authentication, such as SSH, sudo, Active Directory, and other service providers. Additionally, Pass-through Authentication Modules (PAM) are the foundation of the authentication process in Linux and allow for transparent authentication. Using PAM makes it possible to manage multiple authentication methods in a simpler way. There are many MFA tools available for Linux systems, such as Google Authenticator, Duo Security, Azure AD Multi-Factor Authentication, and YubiKey. This tool allows SSH access, sudo access etc on Linux systems. It provides different authentication methods, including push notifications, text messages, phone calls, and hardware tokens that can be used to secure the service. In summary, the recommended Linux authentication method and use of PAM and MFA tools can help increase the security of Linux systems and protect sensitive data from unauthorized access.

11. REFERENCES

[1] A Framework for Secure Linux Based Authentication (Do Wan Thanh, Ivar Jørstad). It addresses an alternative framework which uses mobile based 2FA using PAM.

[2] Building Blocks - Linux PAM Authentication (Lila Zurzolo). This paper showcases the underlying PAM authentication used in Linux systems.

[3] Implementation and Analysis of Two Factor Authentication (Asif Amin, Isar Ul Haq, Monisa Nazir). Analysing the security parameters and efficient implementation of 2FA on systems.

[4] Active Directory and Related Aspects of Security (A. Binduf, H.O. Alamoudi, H. Balahmar). Centralize control of user logins to organization and network systems.

[5] Identity and Access Management as Security as a Service from Clouds (Deepak Sharma, Manish M. Potey). Functionality of Identity Access Management in security as a service.

[6] Identity Management in Linux Systems (Kumar Gunjan, G. Sahoo). Implementation of Identity Access Providers and their use of authentication.

[7] Centralized Authentication using open LDAP (Jokey, Sven Vermeulen). Usage of LDAP for centralized identity provider and fluctual authentication using Ldap instead of linux PAM.

[8] A comparative analysis of HOTP and TOTP (Lina Lumburovska, Stefan Andonov). A deper dive into hash based otp and timed otp.

[9] Advanced Linux Security: A Survey (Matthew R. Yaswinski, MD Minhaz Chowdhury). List of Methods for securing Linux Systems