Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Implementation of Multi-Factor Authentication and One-Time Password for Debian Remote Server Administration

Rani Rosalinda[1]*, Wetina Hulu[2], Lotar Mateus Sinaga[3]

[1,2,3]*Faculty of Computer Science, Informatics Engineering Study Program, Santo Thomas Catholic University Medan*
lindarani786@gmail.com

Abstract

Server security is an important aspect of information technology infrastructure management because servers function as data storage centers and providers of network services. In practice, server administration is commonly performed remotely through the Secure Shell (SSH) protocol. Although SSH provides encrypted communication, authentication based solely on usernames and passwords still has several weaknesses, such as vulnerability to brute force attacks, credential theft, phishing, and unauthorized access. Therefore, an additional security mechanism is needed to strengthen server administrator access protection. This study aims to implement Multi Factor Authentication (MFA) and One Time Password (OTP) for Debian-based remote server administration. The research methods include problem identification, literature review, system design, implementation, testing, and evaluation. The implementation was carried out by integrating Google Authenticator with SSH services through the Pluggable Authentication Module (PAM). The results indicate that MFA and OTP successfully add an extra layer of security to the administrator authentication process. In addition to entering a username and password, users are required to provide a unique OTP code that is valid only for a limited period. Testing results show that the system is capable of rejecting login attempts that fail to meet one of the authentication factors. Therefore, the implementation of MFA and OTP is effective in reducing the risk of unauthorized access caused by password leakage and improving the security of Debian remote server administration.

Keywords: *Debian, server security, Multi Factor Authentication (MFA), One Time Password (OTP), SSH.*

1. Introduction

The development of information technology has encouraged the use of servers as centers for managing data, applications, and network services in various organizations and institutions. Servers play a crucial role in maintaining service availability, making security a key factor. Threats to server security continue to increase as cyberattack techniques develop that target weaknesses in authentication systems and user access management [3], [14].

Server administration is generally performed remotely using the Secure Shell (SSH) protocol. The SSH protocol provides an encrypted communication mechanism so that data sent over the network is more secure than other remote protocols. However, the use of username and password-based authentication still has various weaknesses, such as being vulnerable to brute force attacks, credential theft, phishing, and password leaks [9], [12]. If administrator credentials are successfully obtained by unauthorized parties, then full access to the server can be obtained, potentially causing system damage or leaking important data.

One method widely used to improve authentication security is Multi-Factor Authentication (MFA). MFA is an authentication mechanism that requires users to provide more than one verification factor before gaining access to the system. These factors can be something the user knows (a password), something the user possesses (a smartphone or token), or the user's biometric characteristics [2], [6]. By implementing MFA, the risk of unauthorized access due to password theft can be minimized because attackers must pass through an additional layer of authentication.

MFA implementations generally utilize One-Time Passwords (OTPs), which are unique authentication codes that are only valid for a certain period of time. One widely used OTP method is the Time-Based One-Time Password (TOTP), which generates new codes periodically based on time synchronization between the server and the user's device [8], [15]. The use of OTPs has been proven to increase system security because authentication codes cannot be reused after their validity period has expired.

Several previous studies have shown that implementing MFA is effective in improving system access security. Diaandisy and Risqiwati [1] stated that MFA is able to reduce the risk of brute force attacks on SSH services. Lencana [5] successfully implemented multi-factor authentication on phpMyAdmin to improve administrator access protection. In addition, Haeruddin et al. [6] showed that implementing MFA can optimize data access security in an organizational environment. Another study conducted by Kuswanto and Setiawan [10] also proved that the Two-Factor Authentication (2FA) method is able to improve the security of remote server services compared to conventional password-based authentication.

On the Linux operating system, MFA can be implemented through the Pluggable Authentication Module (PAM), which allows the integration of various authentication methods into system services, including SSH [11]. One widely used application is Google Authenticator because it is free, easy to implement, and supports the TOTP mechanism. Using Google Authenticator allows administrators to obtain OTP codes via registered mobile devices, making the authentication process more secure [13].
Based on these issues, this study aims to implement Multi-Factor Authentication (MFA) using One-Time Password (OTP) based on Google Authenticator on the SSH service of Debian servers. Implementation is carried out through the integration of Google Authenticator with PAM to add a layer of security to the administrator authentication process. This research is expected to improve the security of remote administration of Debian servers and reduce the risk of unauthorized access due to password leaks or theft.

2. Methodology

2.1. Multi-Factor Authentication (MFA)

Multi-Factor Authentication (MFA) is an authentication method used to enhance system security by combining more than one verification factor. These factors typically consist of something the user knows, such as a password, something they own, such as a mobile device, or something associated with the user, such as biometrics. Implementing MFA aims to reduce the risk of unauthorized access due to password theft or leaks, as attackers cannot gain access with just one authentication factor.

2.2. One-Time Password (OTP)

A One-Time Password (OTP) is a unique authentication code that is valid only for a specific period of time or for a single use. OTPs are typically used as an additional layer of security during logins. One commonly used type of OTP is the Time-Based One-Time Password (TOTP), a code that changes periodically , typically every 30 seconds, based on time synchronization between the server and the user's device.

2.3. Secure Shell (SSH)

Secure Shell (SSH) is a network protocol used for remote server access and administration over an encrypted connection. SSH is widely used because it maintains data confidentiality during communication between clients and servers. However, the SSH authentication system, which uses only a username and password, still poses vulnerabilities to attacks such as brute force attacks and credential theft.

2.4. Google Authenticator

Google Authenticator is an application used to generate TOTP-based OTP codes. This application runs on mobile devices and generates codes that change automatically at specific intervals. In this study, Google Authenticator was used to generate additional verification codes during the SSH login process, thereby enhancing user authentication security.

2.5. Pluggable Authentication Module (PAM)

The Pluggable Authentication Module (PAM) is a mechanism in the Linux operating system that allows the integration of various authentication methods into system services. PAM provides the flexibility to add layers of security without requiring changes to the core application. In this study, PAM was used to integrate Google Authenticator into the SSH service on the Debian operating system.

2.6. Debian Operating System

Debian is a Linux-based operating system widely used for servers due to its stability and security. It provides various network services, including a configurable SSH server. In this study, Debian was used as the primary platform for implementing Multi-Factor Authentication (MFA) on the SSH service.

2.7. Research Methods

This research uses an experimental method with a security system implementation and testing approach. This method was used to implement and test the implementation of Multi-Factor Authentication (MFA) and One-Time Password (OTP) in the Secure Shell (SSH) service on a Debian server. The research stages included problem identification, literature review, system design, implementation, testing, and evaluation. During the problem identification stage, an analysis was conducted to identify weaknesses in the SSH authentication system, which still uses usernames and passwords. The literature review phase involved collecting references related to server security, MFA, OTP, SSH, PAM, and Google Authenticator.
Next, in the system design phase, a two-factor authentication mechanism was designed. The implementation phase involved installing and configuring Google Authenticator and integrating it via PAM into the SSH service on Debian. Afterward, testing was conducted using black box testing with several login scenarios to determine the system's success in granting or denying access. The final stage is the evaluation of the results which aims to assess the effectiveness of implementing MFA and OTP in improving the security of administrator access on Debian servers.

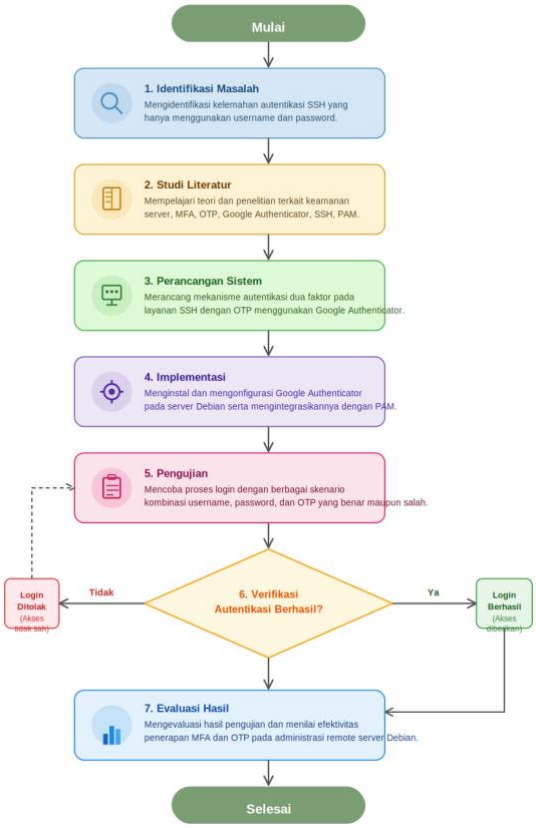

Figure 1: Research method flowchart

Explanation of Research Method Flowchart

1. Problem Identification

At this stage, identification is carried out on weaknesses in SSH authentication which only uses usernames and passwords, thus risking unauthorized access due to credential theft or brute force attacks.

2. Literature Study

This stage is carried out by studying various references related to server security, SSH, Multi Factor Authentication (MFA), One Time Password (OTP), Google Authenticator, and Pluggable Authentication Module (PAM).

3. System Design

At this stage, a two-factor authentication mechanism is designed for the SSH service by adding OTP verification using Google Authenticator as an additional layer of security.

4. Implementation

The implementation stage is carried out by installing and configuring Google Authenticator on the Debian server and integrating it with the SSH service via PAM.

5. Testing

Testing is carried out by trying the login process using several scenarios, such as correct or incorrect combinations of username, password, and OTP to determine system performance.

6. Authentication Verification:

At this stage, the system checks whether all authentication factors have been met. If authentication is successful, access is granted to the user. If any of the authentication factors are not met, access is denied.

7. Evaluation of Results

The final stage is carried out to evaluate the test results and assess the effectiveness of implementing MFA and OTP in improving the security of remote administration of Debian servers.

3. Results

3.1. Implementing Google Authenticator on Debian Server

The implementation phase begins with the installation of the Google Authenticator package on the Debian operating system. The installation process is performed using the apt package manager. After successful installation, Google Authenticator is configured to generate a secret key and QR code for registration on the administrator's device.

The generated QR code is then scanned using the Google Authenticator app on the administrator's smartphone. After successful registration, the app generates a One-Time Password (OTP) that changes every 30 seconds. This OTP is used as a second authentication factor when logging into the server.

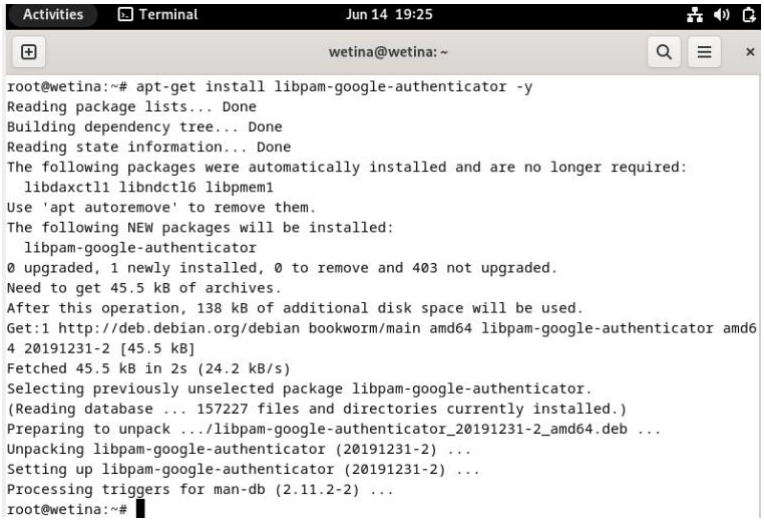

Figure 2: apt-get install libpam-google-authenticator -y

The libpam-google-authenticator package is installed using the apt package manager on the Debian operating system. This package integrates the Google Authenticator service with the Linux authentication system via Pluggable. Authentication Module (PAM), so that the server can support the implementation of two-factor authentication as an additional layer of security.

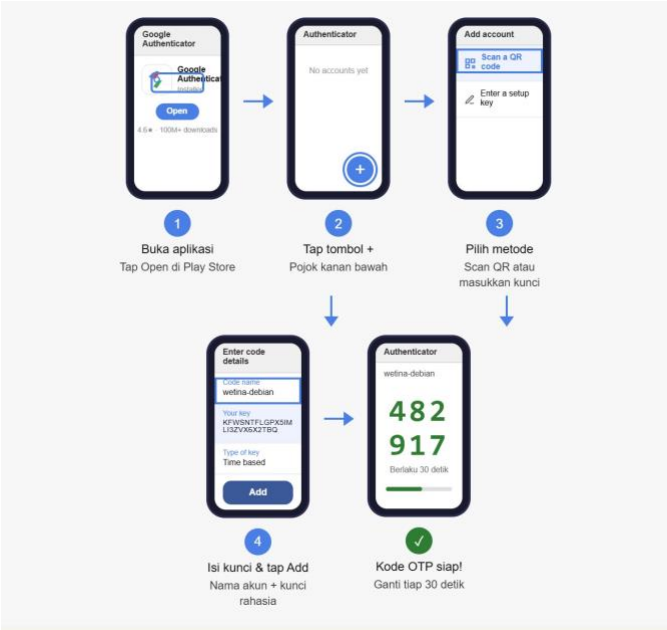

Figure 3: Google Authenticator Configuration Flow for Two-Factor Authentication (2FA)

The configuration process begins by opening the Google Authenticator app and adding a new account using a QR code or setup key obtained from the server. After successful registration, the app will generate a One-Time Password (OTP) that changes automatically every 30 seconds and is used as a second authentication factor when logging into the Debian server.

Figure 4: Google Authenticator QR Code and Secret Key, cat ~/.google_authenticator

The QR code and secret key are used to connect a Google Authenticator account to the Debian server. The QR code can be scanned directly using the Google Authenticator app, while the secret key can be entered manually if scanning is not possible. This information forms the basis for generating the OTP code used for user authentication.

3.2. PAM and SSH Configuration

After Google Authenticator is installed, configure the Pluggable Authentication Module (PAM) and the SSH service. This configuration ensures that the server requests an OTP code after the user enters their username and password. This ensures that the authentication process consists of two layers of security: a password and an OTP. The configuration results indicate that the SSH service has successfully integrated with Google Authenticator. The system can verify the OTP code entered by the user before granting access to the server.

- Code confirmed

```
Your new secret key is: KFW5NTFLGPX5IMLI3ZVX6X2TBQ
Enter code from app (-1 to skip): 746311
Code confirmed
Your emergency scratch codes are:
  19769328
  93583540
  31758630
  29438653
  47267698
```

Figure 5: Google Authenticator Verification (Code Confirmed)

"Code confirmed" status indicates that the Google Authenticator configuration process has been successfully verified by the system. This verification ensures that the secret key used is valid and that the generated OTP code can be used as a second authentication factor for SSH services.

- cat /etc/pam.d/sshd

```
# Standard Un*x password updating.
@include common-password
auth required pam_google_authenticator.so
```

Figure 6: PAM configuration in the /etc/pam.d/sshd file

PAM configuration is achieved by adding the **pam_google_authenticator.so module** to the /etc/pam.d/sshd file. This setting allows the system to request an OTP code generated by Google Authenticator after the user successfully enters their username and password during the SSH login process.

- Configure KbdInteractiveAuthentication on sshd_config (change from no to yes)

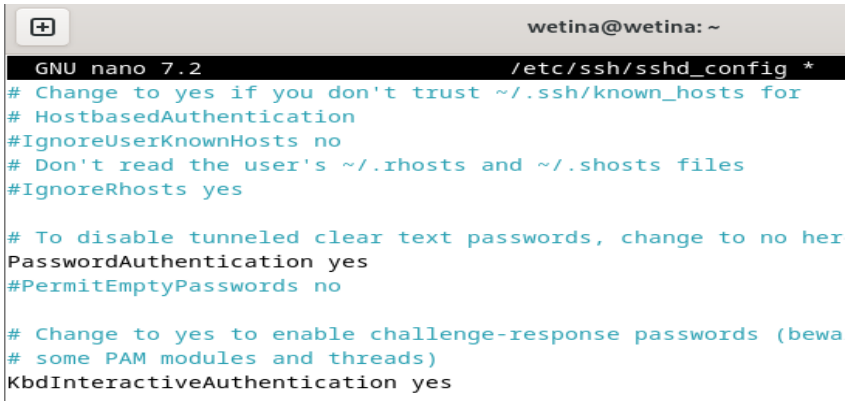

Figure 7: KbdInteractiveAuthentication Configuration in the sshd_config File

KbdInteractiveAuthentication parameter is changed to **yes** in the SSH configuration file (sshd_config). This setting is required for the SSH service to support the interactive authentication mechanism used to verify OTP codes as an additional authentication factor.

3.3. SSH Login Testing

Testing was conducted to determine the success of MFA and OTP implementations in SSH services. Several test scenarios were conducted with different combinations of authentication data.

- **Successful SSH Login with Password and OTP (Scenario 1)**

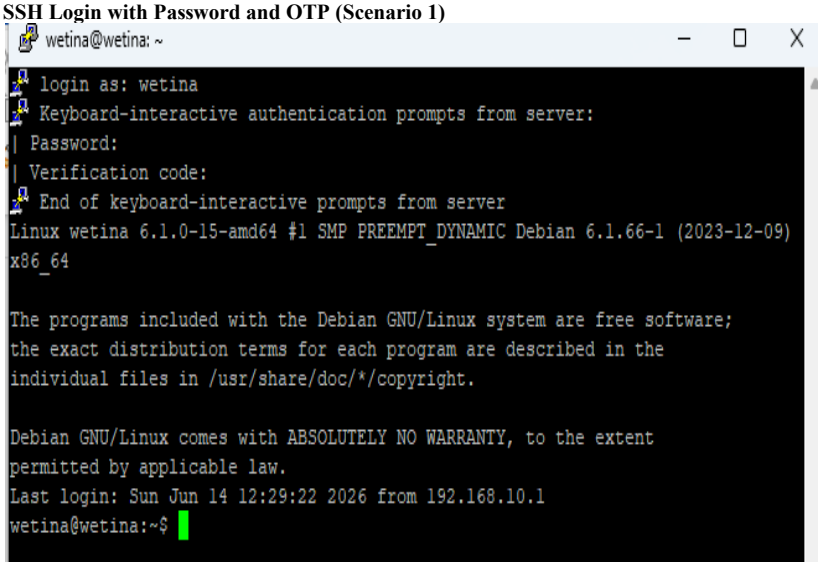

Figure 8: Successful SSH Login Using Password and OTP

The login process was successful after the user entered a valid username, password, and One-Time Password (OTP) code. The system successfully verified all provided authentication factors, granting access to the Debian server. These results demonstrate that the integration of Google Authenticator with the SSH service has run as configured and successfully implemented the Multi-Factor Authentication (MFA) mechanism.

- Scenario 2: Wrong password → Rejected

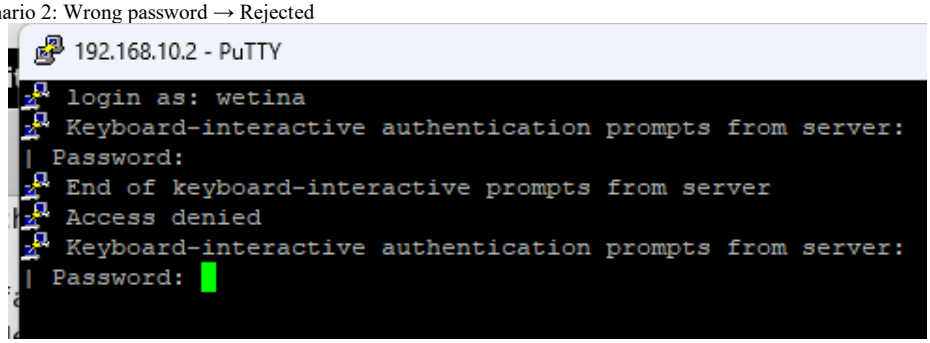

Figure 9: Login Rejected Due to Wrong Password

The login attempt was rejected because the entered password did not match the data stored on the server. Despite the correct username, the system still denied access because one of the key authentication criteria was not met. These test results indicate that the SSH service still verifies the password before proceeding to the OTP authentication stage.

- **Scenario 3: Correct username, correct password, incorrect OTP → Login rejected**

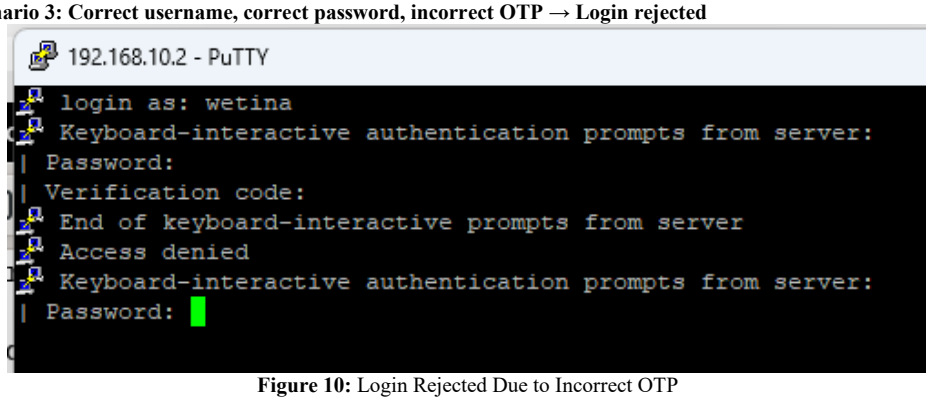

Figure 10: Login Rejected Due to Incorrect OTP

The login process failed because the OTP code entered did not match the code generated by Google Authenticator. Even though the username and password were correct, the system still denied access because the second authentication factor was not met. This situation demonstrates that implementing MFA can provide an additional layer of security against unauthorized access.

- **Scenario 4: Wrong/missing username**

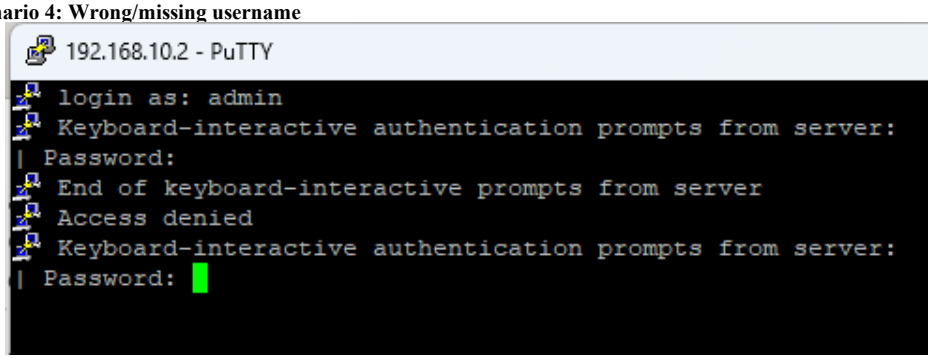

Figure 11: Login Rejected Due to Unregistered Username

The login attempt was rejected because the username used was not registered on the Debian server system. As a result, the authentication process could not proceed to the password or OTP verification stage. This result indicates that the system only grants access to registered users who have the necessary permissions to access the server.

Table 1: SSH Login Test Results Using MFA and OTP

No	Testing Scenario	Results
1	Correct username, correct password, and correct OTP	Successful Login
2	Username is correct, Password is incorrect	Login Denied
3	Username is correct, Password is correct, OTP is incorrect	Login Denied
4	Invalid/missing username	Login Denied

Information:

1. In the first scenario, the system successfully grants access because all authentication factors, namely username, password, and OTP, are entered correctly.
2. In the second scenario, the login is rejected because the password entered does not match even though the username is correct.
3. In the third scenario, the login is rejected because the OTP code entered is incorrect, even though the username and password are correct.
4. In the fourth scenario, the login is rejected because the username used is not registered on the server so the authentication process cannot continue.

Based on testing results, the system only grants access if all authentication factors are entered correctly. If any of the authentication factors are incorrect, the system automatically denies login access.

4. Conclusion

This research successfully implemented Multi-Factor Authentication (MFA) using One-Time Password (OTP) based on Google Authenticator on Debian server SSH service through integration with Pluggable Authentication Module (PAM). Based on the test results, the system only grants access when all authentication factors, namely username, password, and OTP, are fulfilled correctly. The implementation of MFA and OTP is proven to be able to improve the security of remote server administration by providing an additional layer of protection against the risk of unauthorized access due to credential leaks or brute force attacks.

Acknowledgement

We would like to thank our supervisors for their guidance, input, and support throughout this research. We also extend our gratitude to the Computer Science Study Program at Santo Thomas Catholic University, Medan, and all those who have supported us in completing this research successfully.

References

[1] E. Diaandisy and D. Risqiwati, "Analysis and Implementation of Multi-Factor Authentication (MFA) to Prevent Brute Force Attacks on Mikrotik SSH," JIPI (Scientific Journal of Informatics Research and Learning), vol. 10, no. 4, pp. 3872–3885, 2025.
[2] A. Fauzi, I. N. Aprilla, N. Fauziyyah, and N. F. Bachtiar, "Implementation of Multi-Factor Authentication, Single Sign-on and Role-Based Access Control in Information System Security (Literature Review Study)," Fibonacci: Journal of Economics, Management and Finance, vol. 2, no. 1, pp. 33–45, 2025.
[3] M. Huda, Windows & Linux Security Hardening: Best Practices for Data and System Protection, 2025.
[4] A. Imanudin, Proxmox VE-Based Server Virtualization. Excellent Publishing, 2018.
[5] W. B. Lencana, "Implementation of Multi-Factor Authentication on phpMyAdmin," Journal of Information Technology Education and Information Technology, vol. 2, no. 1, pp. 35–39, 2023.
[6] H. Haeruddin, S. E. Prasetyo, and A. Mindy, "Implementation of Multi-Factor Authentication to Optimize Data Access Security at PT. ABC," Journal of Informatics Management (JAMIKA), vol. 15, no. 1, 2025.
[7] D. Ariyus and K. R. Andri, Data Communication. Yogyakarta: Andi Publisher, 2022.
[8] T. S. Gunawan, M. Kartiwi, and N. Ismail, "Implementation of Two-Factor Authentication Using Time-Based One-Time Password for Secure Web Application Access," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 3, pp. 112–119, 2023.
[9] R. Hidayat and S. H. Pramono, "Security Analysis of SSH Protocol Against Brute Force Attacks on Linux Operating System," Journal of Information Technology and Computer Science (JTIIK), vol. 10, no. 2, pp. 245–252, 2023.
[10] D. Kuswanto and B. Setiawan, "Implementation of Network Security Using Two-Factor Authentication Method on Remote Server Services," Journal of Informatics and Electronic Engineering (JIRE), vol. 5, no. 1, pp. 55–63, 2022.
[11] M. Mufadhol and A. Suharto, "Implementation of PAM (Pluggable Authentication Module) to Improve Login Security on Linux Servers," Journal of Business Information Systems (JOSIB), vol. 12, no. 1, pp. 1–9, 2022.
[12] A. Prasetyo, A. Wibowo, and D. Kurniawan, "Comparison of Authentication Methods in SSH Services: Password, Public Key, and Multi-Factor Authentication," National Journal of Electrical Engineering and Information Technology (JNTETI), vol. 12, no. 4, pp. 301–309, 2023.
[13] R. E. Putra and D. P. Lestari, "Evaluation of the Effectiveness of Google Authenticator as a One-Time Password Mechanism in Server Administration Systems," Scientific Journal of Computer Informatics, vol. 29, no. 1, pp. 18–26, 2024.
[14] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. New Jersey: Pearson Education, 2022.
[15] F. Yulianto and R. Rahmat, "Implementation of Two-Factor Authentication in Web-Based Login System Using TOTP (Time-Based One-Time Password)," RESTI Journal (System Engineering and Information Technology), vol. 7, no. 3, pp. 512–520, 2023.