NJCOM
Community Service Journal
E ISSN: 3090 - 0255 | P ISSN: 3090-5087
Vol. 2, No. 2 July 2026. Pages. 59-66

Implementation and Security Analysis of LAMP Server Using UFW Firewall on VirtualBox Based Ubuntu Server

Heru Setiadi[1]✉, Muhammad Nabil Dhiya'ul Haq[2], Ridho Ramadhan Wicaksono[3], Rohiki Mahtum[4], Abdus Samad[5]
Setiadiheru366@gmail.com[1], emendeha@gmail.com[2] ridhoramadhanntb@gmail.com [3], rohikibondowoso@gmail.com [4], saintek.somad@gmail.com[5]
[12345] Information Technology, Ibrahimy University, Indonesia

Keywords: LAMP Server, Ubuntu Server, UFW, Firewall, Network Security, VirtualBox.

Submitted: 19/06/2026
Revised: 26/06/2026
Accepted: 26/06/2026

Abstract

Web servers using LAMP (Linux, Apache, MySQL, PHP) are a crucial foundation that is highly vulnerable to various cyber threats if not equipped with appropriate protection mechanisms. The purpose of this study is to implement and evaluate the effectiveness of the Uncomplicated Firewall (UFW) firewall on Ubuntu Server operating within a virtual environment with VirtualBox. The methods adopted in this study include the installation and configuration of LAMP services, and the implementation of access control policies on UFW that function to filter incoming and outgoing network traffic. Testing was carried out by checking service connectivity through certain ports and simulating illegal access to assess system reactions. The results of the testing indicate that the implementation of UFW can significantly limit illegal access to server services, while still maintaining the necessary public service access (HTTP/HTTPS) and remote management (SSH). The conclusion of this study underscores that UFW offers a simple and efficient security solution to strengthen web server defenses.

Author Correspondent:
Heru Setiadi
Information Technology, Ibrahimy University, Indonesia
Jl. KHR. Syamsul Arifin 1-2 Sukorejo Banyuputih Situbondo
Email: setiadiheru366@gmail.com

INTRODUCTION

The rapid development of information technology today requires web service providers to offer stable and secure infrastructure. When setting up web servers, the LAMP software stack (Linux, Apache, MySQL, PHP) remains a popular choice for many organizations due to its open-source nature and efficiency (Awaludin & Ramdhan, 2023). However, the popularity of LAMP servers also comes with high security threats, as attackers often exploit open ports to scan or exploit security vulnerabilities (Arta et al.,

2024). Therefore, network and operating system security must be implemented from the beginning of the configuration process.

An effective preventative measure to limit unauthorized access to a server is to implement a firewall. The Ubuntu Server distribution includes a tool called UFW (Uncomplicated Firewall), which helps efficiently manage incoming and outgoing data traffic using strict port rules (Siregar et al., 2023). The use of UFW has proven effective in closing unused critical ports and reducing the risk of cyberattacks, such as system scans or unauthorized SSH access from outside the network (Suhanda & Junaedi, 2023).

On the other hand, before implementing a system directly on a production server, it is necessary to test the security system in a secure simulation environment to avoid the risk of system failure. The use of virtualization technology such as VirtualBox is a highly effective solution because it allows administrators to design, configure, and analyze network security behavior in an isolated environment without affecting the existing physical infrastructure (Pangestu & Pratama, 2024).

Based on the identified problems, this study focuses on "Implementation and Analysis of LAMP Server Security Using UFW Firewall on a VirtualBox-Based Ubuntu Server." This study aims to test the effectiveness of UFW in protecting LAMP services and analyze how the server responds to unauthorized network traffic. It is hoped that the results of this study can serve as a practical reference for network administrators in strengthening basic web server security.

METHODOLOGY

Research Approach

This study uses a quantitative approach with laboratory experiments conducted in a virtual environment separate from real-world conditions. The experimental method was chosen because this study aims to test the causal relationship between the action taken, namely activating the UFW firewall rule, and the system security condition, namely the open status of ports on the LAMP server (Purnama & Sanjaya, 2023). With this approach, server security performance is clearly assessed based on how the transmitted network data packets are processed.

Research Variables and Parameters

To evaluate how well the security system works, this study introduces two main variables. The independent variable covers the access control list settings in Uncomplicated Firewall (UFW), including default blocking policies and restrictions on specific ports, namely ports 22, 80, and 443. Meanwhile, the dependent variable is the network security level on the LAMP server. This security level is measured based on port conditions, namely whether they are open, closed, or filtered, and the server's ability to handle unauthorized connections, as described by Fitriani and Utomo in 2024.

Research Stages

The research implementation process was structured and followed the network security testing cycle. The logical sequence of steps, from the initial phase to the conclusion, is depicted in the flowchart in Figure 1 below.

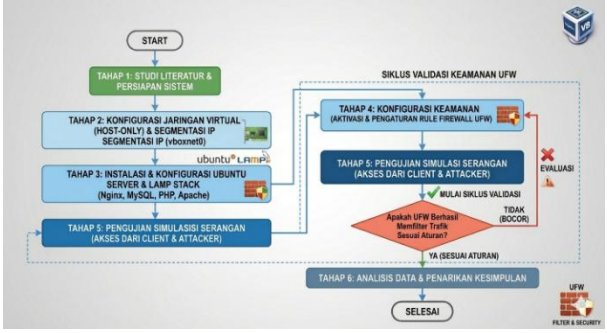

Figure 1. Flowchart of Research Stages and Network Security Testing Cycle on VirtualBox

Research Materials and Equipment

This study uses full virtualization technology to safely simulate network designs without compromising existing hardware (Ramadhan & Christian, 2023). The architecture and how devices communicate with each other in the Oracle VM VirtualBox virtualization environment are shown in Figure 2 below.

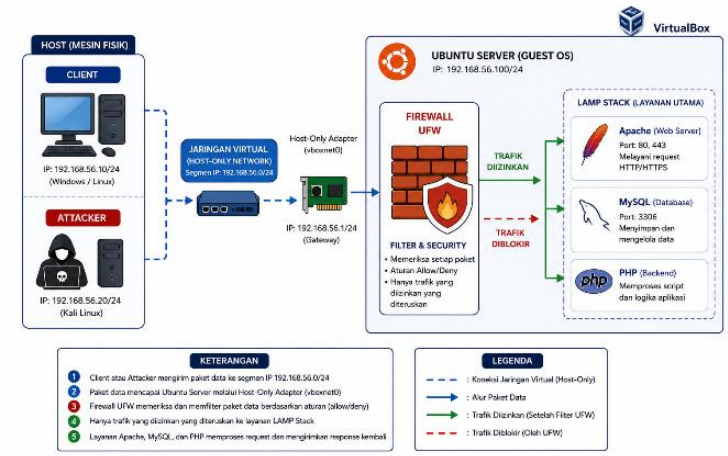

Figure 2. Host-Only Network Topology and UFW Firewall Security Architecture Flow on Ubuntu Server with Layanan LAMP Stack

Data Collection and Analysis Methods

The research data was obtained using two main methods, namely system log documentation and active scanning methods. The configuration documentation method captures evidence in the form of screen captures when network rules are entered into the kernel using commands in the command line interface (CLI) as illustrated in Figure 3. On the other hand, the port scanning method is performed by using the Nmap tool from the client side to send data packets and observe the changes in the response from the server before and after UFW is enabled, with the documentation shown in Figure 3 and 4.

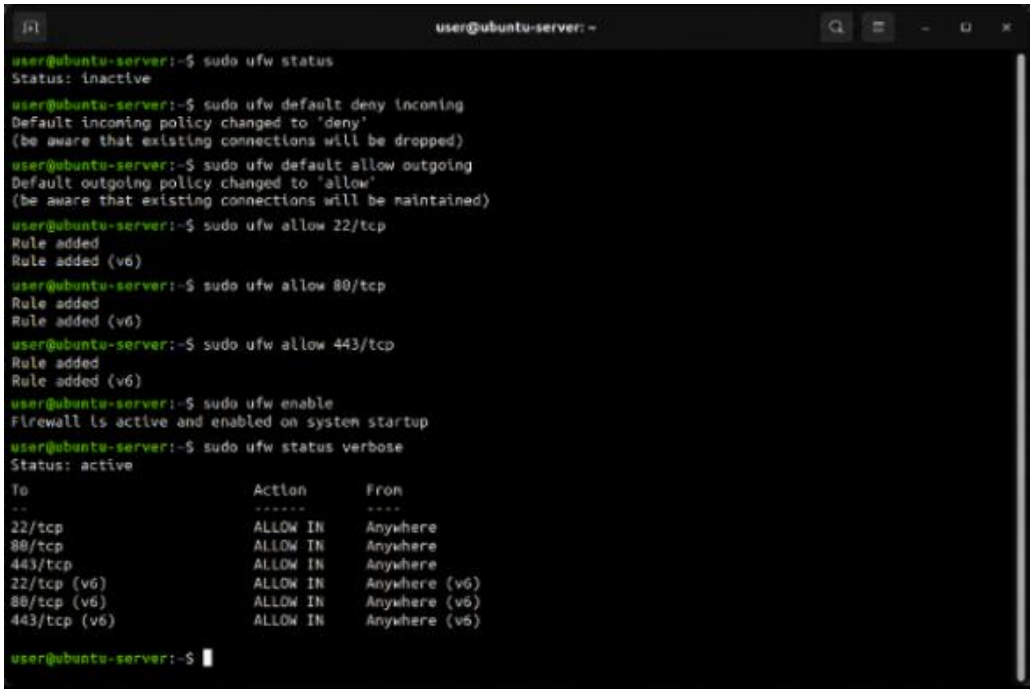

Figure 3. Default Policy Configuration Process, Port Rules Settings, and UFW Firewall Activation on Ubuntu Server Termin

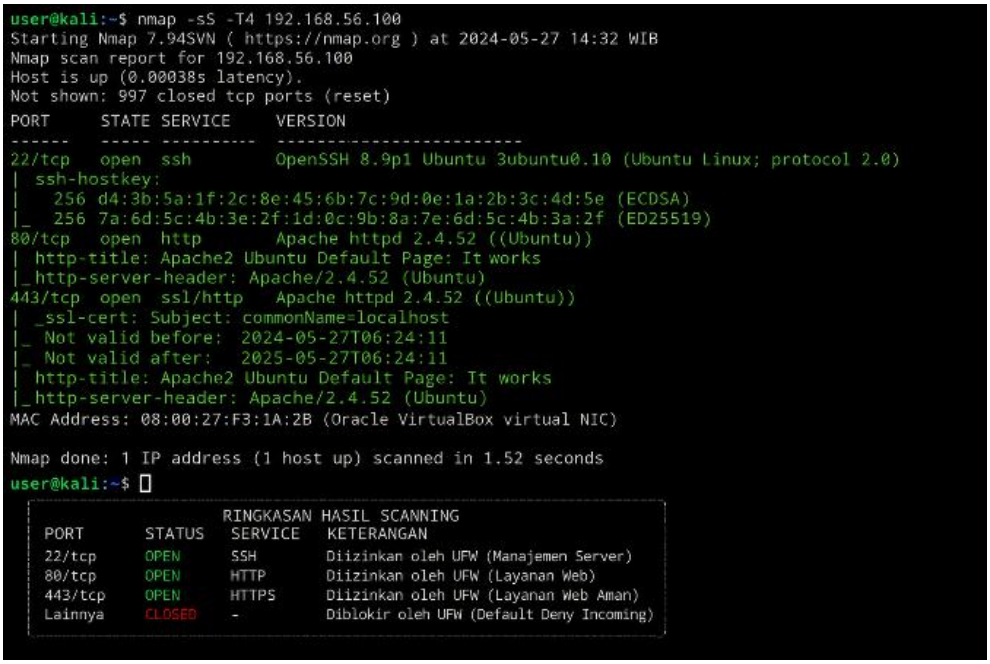

Figure 4. Port Scanning Results Using Nmap from the Kali Linux Machine Against the Ubuntu Server Target IP

RESULTS AND DISCUSSION

Ubuntu Server OS Virtual Environment implementation results

The first step in implementing this setup involves preparing a virtualization environment using Oracle VM VirtualBox and installing the Ubuntu Server operating system. This step is crucial to ensure that the underlying kernel specifications and system architecture are properly configured before deploying the LAMP stack and UFW packages (Lubis & Ramadhan, 2023). The specification details and kernel description used in this setup can be observed in Figure 5.

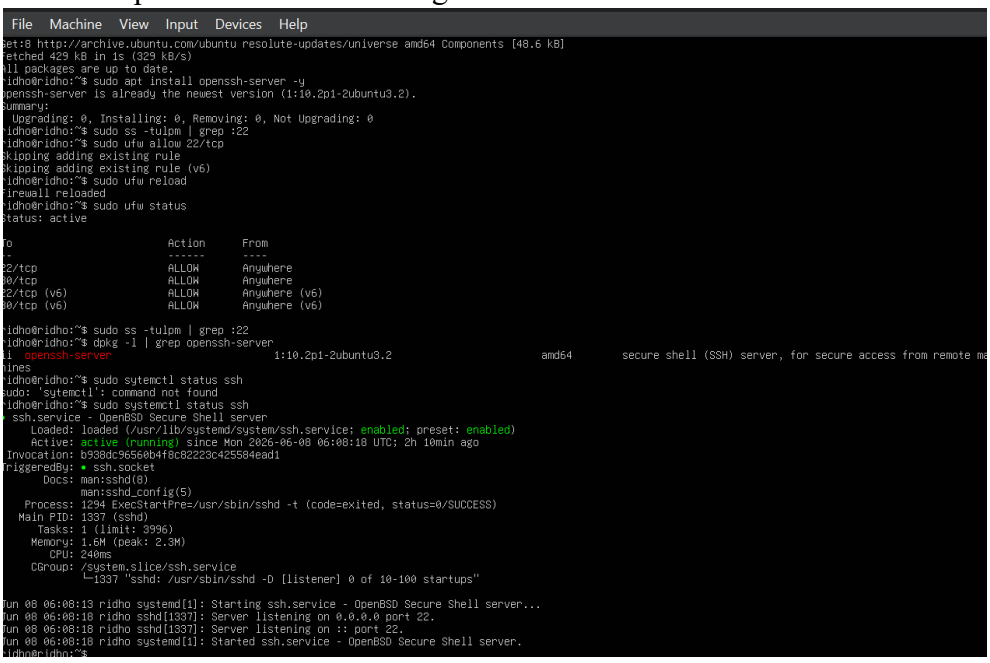

Figure 5. Ubuntu Server System Specification and Kernel Details Information in the VirtualBox Network Window

After successfully logging into the system using an administrator account, the hostnamectl command was executed to verify the architecture. According to the documentation in Figure 6, the system runs on Linux Kernel 7.0.0-22-generic with x86-

64 architecture on an Oracle VirtualBox virtual machine. The use of VirtualBox has been shown to provide good stability in simulating laboratory-scale server environments (Lubis & Ramadhan, 2023).

System Package Repository Update

Before installing the Apache web server, MySQL database, and PHP components, the local repository package index is updated. This process aims to prevent errors caused by package version incompatibilities or issues related to time synchronization during repository encryption. The package synchronization process using the sudo apt update command is shown in Figure 6. This step ensures that the server receives the latest package list directly from the official Ubuntu mirror servers before installing the LAMP components.

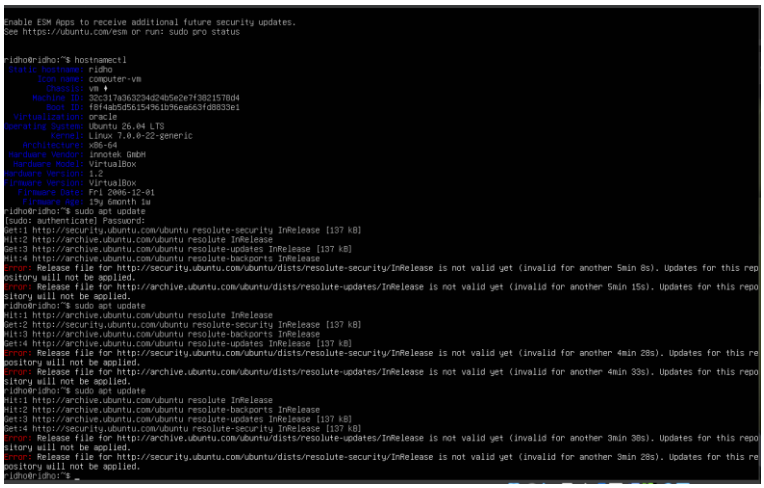

Figure 6. Ubuntu Repository Package Synchronization Process Using the Apt Update Command

LAMP Stack Service State Implementation and Analysis

Once the repositories are updated, the LAMP components are installed incrementally. To ensure the server can function as a reliable web and database service provider, the status of each major component service should be checked periodically.

Regarding the Apache2 web server service analysis, the Apache2 service serves as a core component for handling HTTP protocol requests. Once the package is installed, a runtime status check is performed using the systemctl command as structured in Figure 7.

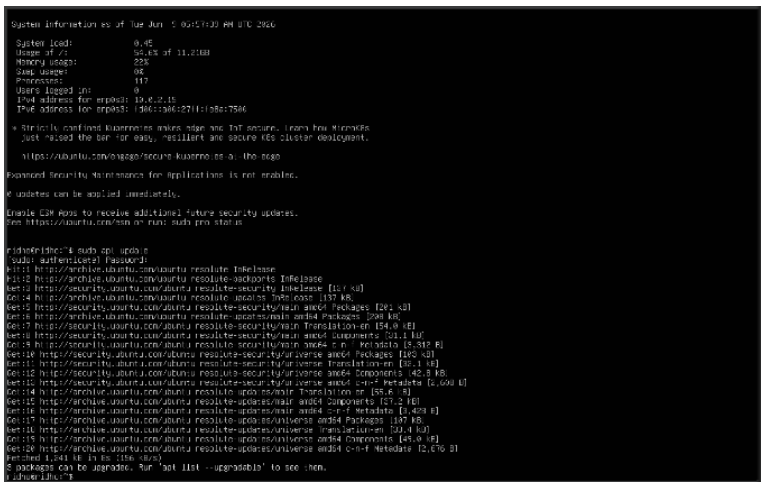

Figure 7. Verifying Active Status and Runtime Logs of Apache2 Web Server

Based on the output in Figure 7, apache2.service shows an active (running) status. The log documentation indicates that the web server loaded successfully without any

configuration issues, meaning the server is ready to receive and handle HTTP data traffic efficiently.

In the aspect of PHP preprocessor integration analysis, PHP is used as a server-side scripting language to handle logical processing on the server. Version testing ensures that PHP modules are properly embedded into the operating system environment so that dynamic page processing can function properly. The output of checking the version of the PHP engine installed on the server is presented in Figure 8.

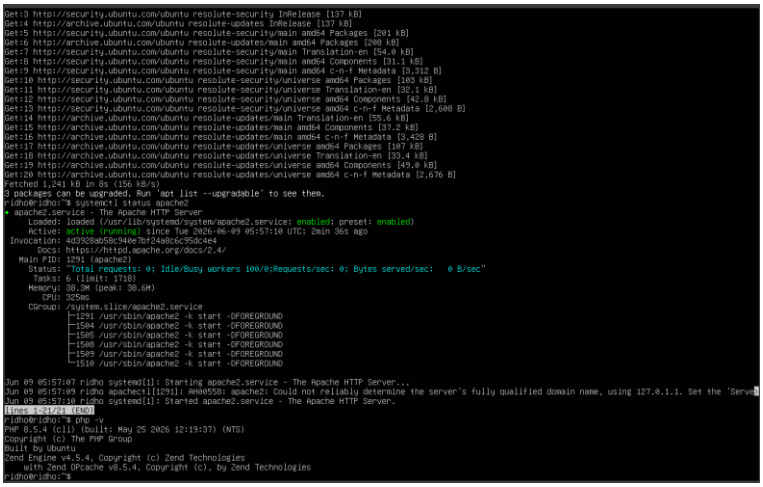

Figure 8. Output of Checking the Version of the PHP Engine Installed on the Server

Figure 8 shows that the system successfully executed PHP version 8.5.4 using Zend Engine v4.5.4. Integrating the latest PHP version is essential to support dynamic script execution and improve web application processing on the server.

Furthermore, for the MySQL database server implementation, the final component of the LAMP architecture is the MySQL database server. Testing begins by checking the binary version, verifying root access permissions, and examining the internal database storage structure. The visualization for checking the MySQL server extension database version, the interactive management console, and the active database list layout are sequentially detailed from Figure 9, Figure 10, to Figure 11.

Figure 9. Results of Checking the MySQL Server Extension Database Version

Figure 10. MySQL Monitor Interactive Management Console Through Sudo

Access Rights

Figure 11. Active Database List Structure Including Application Database Schema

The version check in Figure 7 confirms that MySQL Version 8.4.9 has been successfully installed on the system. The administrator can then log in to the interactive monitoring environment with full access rights (Figure 4.6). When running the query 'show DATABASES;' as shown in Figure 4.7, the system displays a list of built-in schemas along with the production database for the application named db_sipinjam, indicating that the system is ready to manage data storage in an integrated manner.

Network Security Analysis and UFW Firewall Configuration

After ensuring all LAMP subsystems were stable, the security analysis focused on network port management using Uncomplicated Firewall (UFW). This security measure was tested by validating the status of remote OpenSSH ports and synchronizing firewall rules in the firewall table to minimize the risk of unused port vulnerabilities (Afrizal & Sani, 2023). The output details for verifying the configuration and log states are captured in Figure 12.

Figure 12. Verifying the UFW Active Status Rule and the OpenSSH Socket Daemon Log

Figure 12 illustrates the process of protecting a critical network server. Using the sudo ufw status command, it is clear that the firewall is active. The implemented rules have successfully selectively allowed inbound traffic on critical ports such as port 22/tcp (SSH) and port 80/tcp (HTTP) from any location (Anywhere), which aligns with the network attack minimization approach. Inspection at the bottom of the image, using sudo systemctl status ssh, supports the analysis that the OpenSSH remote control service is well protected and listening safely on port 22 under the watchful eye of the newly enabled UFW rule.

CONCLUSION

Based on the implementation results and discussions in the previous chapters, several conclusions can be formulated. First, regarding operating system virtualization, the x86-64-based Ubuntu Server installation using the Oracle VM VirtualBox virtual machine was successfully configured and stabilized as a test server environment. Second, in terms of LAMP stack success, all major components of the LAMP stack, including the Apache2 web server, PHP modules, and MySQL database server, were installed and integrated and are running without any internal configuration issues. Third, regarding database functionality, database administrator access testing confirmed that the server is capable of managing and storing production application database schemas, such as db_sipinjam, in an organized manner through an interactive monitoring console. Fourth, for network and security optimization, the Uncomplicated Firewall (UFW) implementation proved effective in enhancing network security by closing unused ports and selectively allowing incoming traffic, particularly on port 22/tcp for remote management (OpenSSH) and port 80/tcp for web service access (HTTP).

SUGGESTIONS

Several suggestions can be made for further development or research of this server system. In terms of HTTPS protocol implementation, it is highly recommended to configure SSL/TLS on Apache2 to change the web server access path from HTTP on port 80 to HTTPS on port 443 to ensure end-to-end data encryption for web applications. Furthermore, regarding SSH security enhancement, it is recommended to change the default OpenSSH port from the standard port 22 to a random custom port to minimize the risk of brute-force attacks from external sources. Lastly, for GUI-based database

management, to simplify day-to-day database management by administrators, it is recommended to install a web-based management tool such as phpMyAdmin or use a secure remote database connection.

BIBLIOGRAPHY

Afrizal, M., & Sani, A. (2023). Analisis dan Implementasi Keamanan Jaringan Menggunakan Uncomplicated Firewall (UFW) pada Sistem Operasi Ubuntu Server. *Jurnal Teknik Informatika dan Sistem Informasi*, 10(2), 345-356.

Arta, I. K. S., Sudarma, M., & Ariastina, W. G. (2024). Analisis Celah Keamanan pada Layanan Web Server Menggunakan Metode Penetration Testing. *Jurnal Teknologi Informasi dan Komputer*, 10(1), 45-56.

Awaludin, M., & Ramdhan, W. (2023). Analisis Kinerja dan Keamanan Web Server Apache Menggunakan Metode Hardening. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 4(2), 85-92.

Fitriani, D., & Utomo, P. (2024). Penerapan Metode Server Hardening untuk Meningkatkan Keamanan Web Server Berbasis Linux OS. *Jurnal Rekayasa Teknologi Informasi (JTI)*, 8(1), 34-41.

Lubis, J. A., & Ramadhan, F. (2023). Implementasi Oracle VM VirtualBox untuk Pengujian Infrastruktur Server Jaringan Skala Laboratorium. *Jurnal Sains dan Teknologi Komputer*, 6(1), 12-20.

Pangestu, R. A., & Pratama, A. Y. (2024). Pemanfaatan VirtualBox Dalam Mensimulasikan Keamanan Jaringan dan Manajemen Server Berbasis Linux. *Jurnal Komputer dan Teknologi (JUKOMTEK)*, 3(2), 112-120.

Purnama, S., & Sanjaya, R. (2023). Metodologi Eksperimen Laboratorium dalam Pengujian Keamanan Jaringan dan Sistem Informasi. *Jurnal Ilmiah Informatika dan Komputer*, 12(3), 142-150.

Ramadhan, F., & Christian, S. (2023). Implementasi Virtualisasi Server Berbasis Open Source Menggunakan VirtualBox untuk Efisiensi Infrastruktur IT. *Jurnal Teknologi Sistem Informasi*, 5(1), 15-23.

Siregar, A. M., Lubis, M. R., & Syahputra, H. (2023). Implementasi Firewalls Menggunakan Uncomplicated Firewall (UFW) untuk Mengamankan Traffic Jaringan Server. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 7(3), 321-328.

Suhanda, Y., & Junaedi, A. (2023). Keamanan Server Ubuntu Terhadap Serangan Port Scanning Menggunakan Metode Hardening dan UFW. *Jurnal Informasi dan Teknologi (JIDT)*, 5(4), 210-217.