ISOMAse International Society of Ocean, Mechanical and Aerospace Scientists and Engineers

Journal of Ocean, Mechanical and Aerospace
-Science and Engineering-
30th November 2025. Vol.69 No.3

DOI:
10.36842/jomase.v69i3.556

Implementation of a Computer Network Security System Using the Random Port Knocking Method on the Linux Operating System at Insan Prima Mandiri Vocational School

Rifki Nurpalah [a,*], Moch Zenal Zaelani [a], Khaulyca Arva Artemysia [a] and Fikri Fahru Roji [a]

[a)]*Department of Electrical Engineering, Engineering Faculty, Universitas Garut, Indonesia*

*Corresponding author: rifki.nurpalah@uniga.ac.id

Paper History

Received: 25-August-2025
Received in revised form: 28-November-2025
Accepted: 30-November-2025

ABSTRACT

Technological advancements encourage schools to utilize computer networks for learning, making information security crucial. The purpose of this research is the implementation of a computer network security system using the random port knocking method on the linux operating system at Insan Prima Mandiri Vocational School. SMK Insan Prima Mandiri faces the risk of cyber attacks on local and wireless networks. To address this, a server security system was built using a honeypot, port knocking, and iptables. Tests were conducted before and after the system was implemented using port scanning and brute-force attacks. Results showed that before the system was implemented, the SSH (Secure Shell) port was easily accessible. After implementation, the server was able to detect and block attacks, redirect access to the honeypot, and send notifications to the admin via Telegram.

KEYWORDS: *Honeypot, IPTABLES, Port Knocking.*

1. INTRODUCTION

The rapid development of information technology has brought significant impacts to various aspects of life, including the educational sector. The utilization of computer networks in school environments has become an essential requirement to support teaching-learning activities and data management. However, conversely, the increasing network connectivity has also generated threats to information security, particularly cyber attacks such as port scanning and brute force attacks targeting school network systems [1].

One of common vulnerability frequently exploited by attackers is the presence of open ports such as SSH (Secure Shell), which enables remote control access to servers. Although firewalls such as IPTABLES on Linux operating systems have been employed to protect networks, conventional firewalls possess limitations in dynamically recognizing legitimate users. Therefore, additional mechanisms that are more adaptive are required to enhance network resilience against potential intrusions [2].

This research proposes the implementation of a random port knocking method as an additional authentication system to secure access to specific ports. The combination of this method with IPTABLES firewall, Cowrie honeypot as a decoy server, and real-time notification system through Telegram is expected to significantly enhance server security. The implementation was conducted at SMK Insan Prima Mandiri as a case study, with testing performed on the system's effectiveness in confronting real attack scenarios [3].

The research employed a rigorous experimental approach to assess the defensive capabilities of the newly implemented system. This validation involved a pre- and post-implementation comparison of system performance against two prevalent and distinct attack vectors: port scanning and brute force attacks. The selection of these attack types provides a comprehensive test of the system's ability to withstand both reconnaissance activities (port scanning) and direct credential compromise attempts (brute force). The quantitative results successfully demonstrate the system's robust security improvements, highlighting its tripartite functionality.

The ability to conceal critical network ports from attackers, the mechanism for automatically blocking malicious IP addresses, and the capacity to redirect unauthorized access seamlessly to a honeypot environment. The findings confirm the system's practical capability to mitigate immediate and persistent threats by effectively disrupting the attacker's kill chain from the reconnaissance phase through to the exploitation phase. By successfully masking assets and isolating threats via the honeypot, the system contributes significantly to the field of active cyber defense. This established effectiveness provides a strong foundation for future research focused on extending the system's capabilities, such as integrating advanced threat intelligence for predictive defense or applying machine learning algorithms to enhance the autonomous decision-making processes for IP blocking and redirection [4-5].

JOMAse | Received: 25-August-2025 | Accepted: 30-November-2025 | [(69) 3: 264-269]
Published by International Society of Ocean, Mechanical and Aerospace Scientists and Engineers, www.isomase.org., ISSN: 2354-7065 & e-ISSN: 2527-6085

ISOMAse International Society of Ocean, Mechanical and Aerospace Scientists and Engineers
Journal of Ocean, Mechanical and Aerospace
-Science and Engineering-
30th November 2025. Vol.69 No.3
DOI: 10.36842/jomase.v69i3.556

To strengthen the research, an experimental approach was utilized by comparing conditions before and after the system implementation against two common types of attacks: port scanning and brute force attacks. The results demonstrate that the system is capable of concealing critical ports, blocking attacker IPs, and automatically redirecting access to the honeypot [4].

The findings of this study emphasize necessity of implementing layered security mechanisms in educational institutions, where limited resources must be balanced against the growing risks of cyber threats. By integrating random port knocking with existing firewall rules, honeypot deployment, and automated alert systems, the proposed solution offers a cost-effective yet reliable defense strategy [5].

2. METHODS

The methodology employed in this research comprises three principal stages: system design, system implementation, and system performance testing and evaluation. All stages were conducted to realize a random port knocking-based network security system implemented on Linux servers within the SMK Insan Prima Mandiri environment.

2.1 System Design

This research commenced with the design phase of a random port knocking method-based network security system The design was undertaken to provide a comprehensive overview regarding the system workflow and interactions between users, servers, and security components. This system was designed considering three fundamental elements: input, process, and output, as illustrated in Figure 1. Input originates from clients performing sequential port knocking to access SSH services. The processes occurring within the system include knock sequence verification, IP address validation, and security command execution. Should the knocking sequence be incorrect or suspicious login attempts occur, the system will perform IP blocking, randomly alter the knock sequence, and transmit notifications via Telegram. The system output comprises information to administrators regarding network security status and redirection of malicious connections to the honeypot [6].

To clarify the system workflow, Figure 2 was created in the form of a flowchart. This flowchart illustrates the system logic when detecting knocks from clients, verifying access, and handling cases of failed login attempts exceeding three times. Under such conditions, the system will automatically change the knock sequence, block the attacker's IP address, and redirect the connection to a decoy server that has been configured through the Cowrie honeypot. This process enables administrators to monitor attacker activities passively [7].

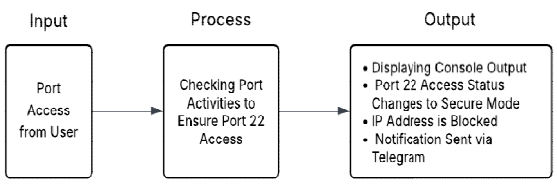

Figure 1: Network security system block diagram

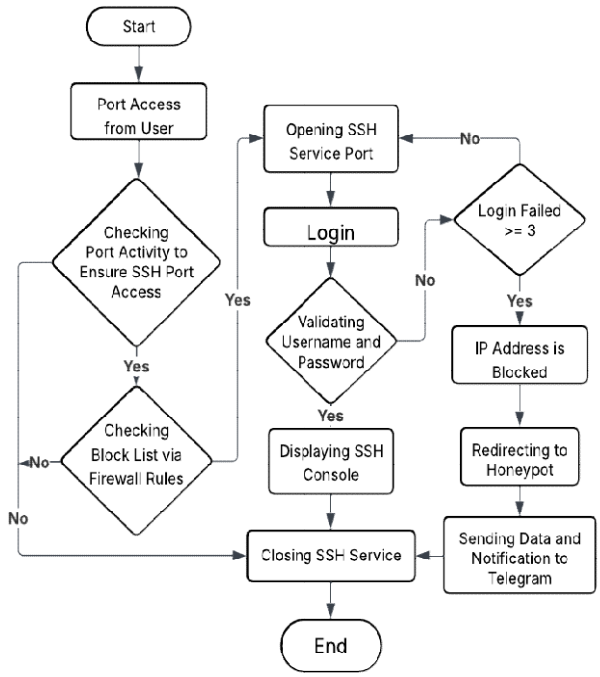

Figure 2: Security system flowchart

The network structure utilized in this research is explained through Figure 3, which displays the network topology at SMK Insan Prima Mandiri. The topology demonstrates that the server and client are located within the same local network, connected through a router. The server functions as the center for security system testing, while the client is employed to conduct attack simulations. The IP address information used in each device is explained in detail in Table 1, which contains the IP configuration of the server and client to support the communication process within the network [8].

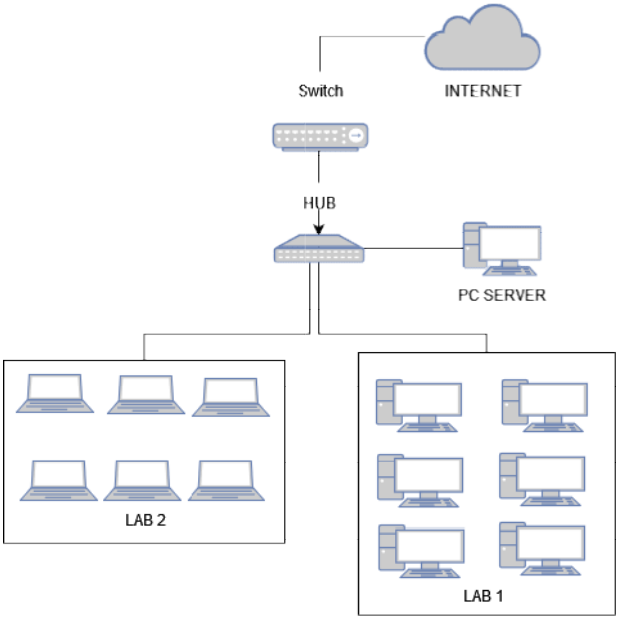

Figure 3: Network topology of Insan Prima Mandiri Vocational School

JOMAse | Received: 25-August-2025 | Accepted: 30-November-2025 | [(69) 3: 264-269]
Published by International Society of Ocean, Mechanical and Aerospace Scientists and Engineers, www.isomase.org., ISSN: 2354-7065 & e-ISSN: 2527-6085

2.2 System Implementation

The implementation phase was conducted by preparing hardware and software that support the deployment of the security system. The hardware consists of one server computer unit with Intel Xeon processor specifications, 8 GB RAM, and 1 TB HDD; one client computer unit with Intel Core i3 processor specifications and 4 GB RAM; and a router to connect both devices [9].

Table 1: IP used SMK Insan Prima Mandiri

No	Network devices	Port Ethernet	IP Address
1	Server	Ether 0	Dynamic Host Configuration Protocol (DHCP)
			192.168.1.1/24
2	PC-Client	Ether 0	Dynamic Host Configuration Protocol (DHCP)
		Ether 1	-
3	Client connected to LAN	Port Switch	Dynamic Host Configuration Protocol (DHCP)

From the software perspective, the Linux Ubuntu operating system was employed on the server as the primary testing platform, while Kali Linux was utilized on the client to conduct attack simulations. Several additional applications were also implemented, including Knockd as a knock sequence detection daemon, IPTABLES as a firewall to block or permit access based on specific rules, and Fail2ban to detect and respond to failed SSH login attempts. Furthermore, Cowrie honeypot was installed to capture and record redirected attacker activities, and Visual Studio Code was used to write Python programs responsible for randomizing knock sequences and automatically sending notifications to Telegram [10].

The system implementation was carried out by developing Python scripts capable of generating random knock sequence combinations and updating the configuration file /etc/knockd.conf. When suspicious login attempts are detected, the Python program will automatically change the knock sequence, restart the knockd service, and send the new sequence to administrators via Telegram Bot. The Cowrie honeypot was then configured to run on port 22. Through rule redirection in IPTABLES, all connections from blocked IPs will be directed to the honeypot port, enabling malicious activities to be observed and recorded without disrupting the main system [11].

3. RESULTS AND DISCUSSION

After the random port knocking-based network security system was successfully designed and implemented on the Linux Ubuntu server at SMK Insan Prima Mandiri, a series of tests were conducted to evaluate the system's effectiveness in confronting port scanning and brute force attack threats. Testing was performed under two conditions: before system implementation and after system implementation, to determine changes in network security levels [12].

3.1 Port Scanning Test Results

Port scanning testing was conducted using the Nmap tool executed from the client computer to detect open ports on the server. In the initial condition before system implementation, scanning results indicated that port 22 (the default port for SSH) was in an open state, potentially creating vulnerabilities for attackers to perform unauthorized access. This is visually displayed in Figure 4, which shows scan results with "open" status on port 22 [13].

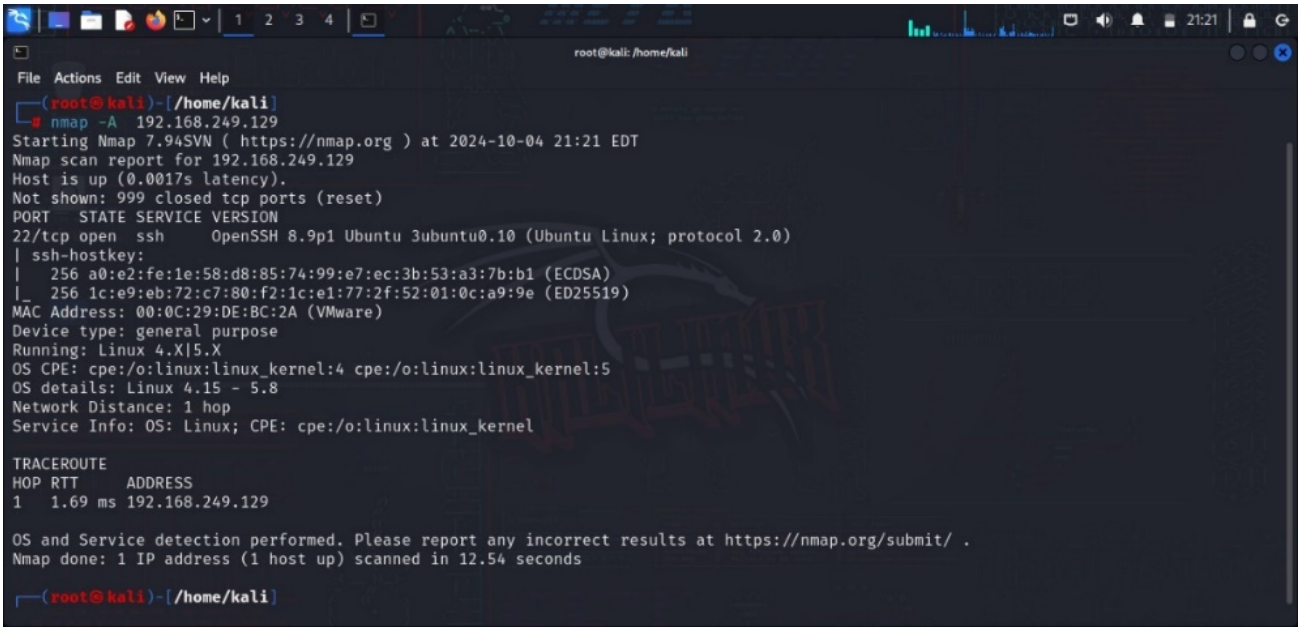

Figure 4: Port scanning test results before system implementation

JOMAse | Received: 25-August-2025 | Accepted: 30-November-2025 | [(69) 3: 264-269]
Published by International Society of Ocean, Mechanical and Aerospace Scientists and Engineers, www.isomase.org., ISSN: 2354-7065 & e-ISSN: 2527-6085

ISOMAse
International Society of Ocean, Mechanical and Aerospace Scientists and Engineers
Journal of Ocean, Mechanical and Aerospace
-Science and Engineering-
30th November 2025. Vol.69 No.3
DOI:
10.36842/jomase.v69i3.556

After the system was implemented, where only IPs performing knocks with the correct sequence were permitted to access port 22, rescanning was conducted using Nmap. The results indicated that port 22 was not detected by the scanner and had a "filtered" status. This demonstrates that the system successfully concealed the SSH port from external scanning. The visualization of these results is shown in Figure 5, which illustrates that port 22 does not appear in the list of ports readable by Nmap. These results indicate that the IPTABLES firewall successfully restricted access to the SSH port dynamically according to the predetermined knock sequence [14].

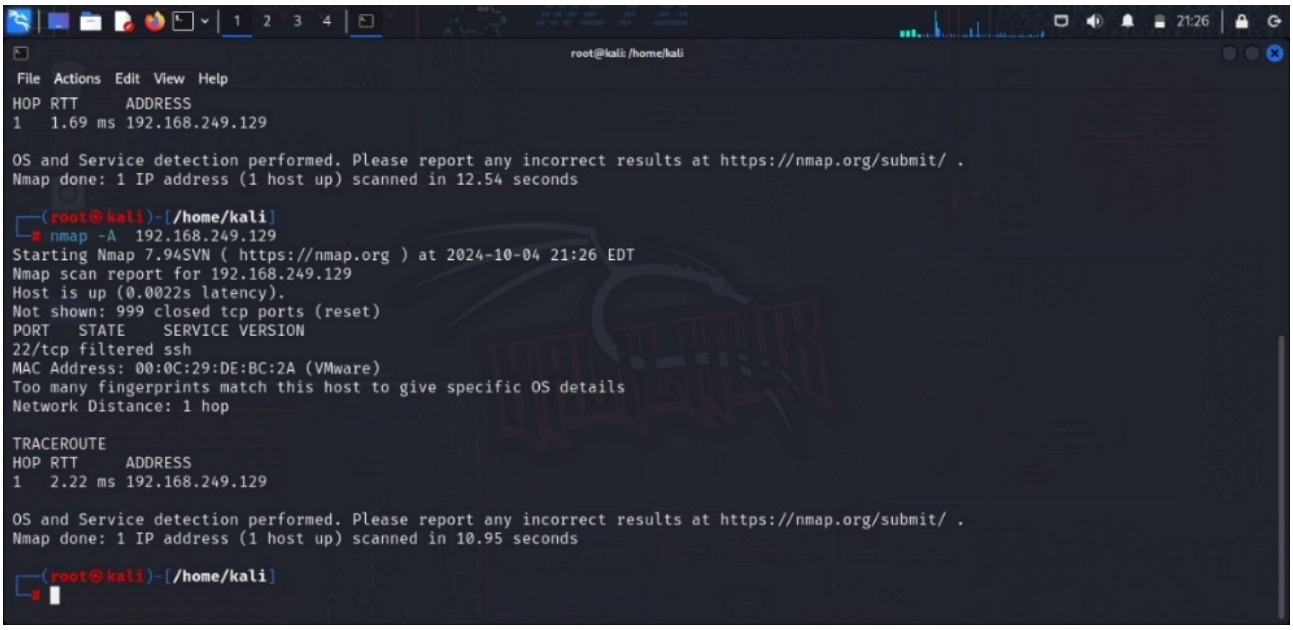

Figure 5: Port scanning test results after the system is implemented

3.2 Brute Force Attack Test Results

Subsequently, testing was conducted against brute force attacks using the Hydra tool. In the initial testing before system implementation, Hydra successfully performed SSH login attempts by matching usernames and passwords through dictionary attack methods. This demonstrates that without additional protection, the server is vulnerable to automated login attacks.

After system implementation, retesting was conducted with the same procedures. However, the system configured with Fail2ban detected three failed login attempts, and then automatically blocked the attacker's IP. Additionally, the integrated Python script would randomly change the knock sequence, update the knockd.conf configuration, and send notifications to administrators via Telegram. Figures 6 and 7 illustrate the activity observed during this process. Figure 6 specifically displays the resulting blocking and connection redirection, showing the system's defensive actions. Meanwhile, Figure 7 provides a view of the attacker's actions via the Cowrie honeypot activity logs.

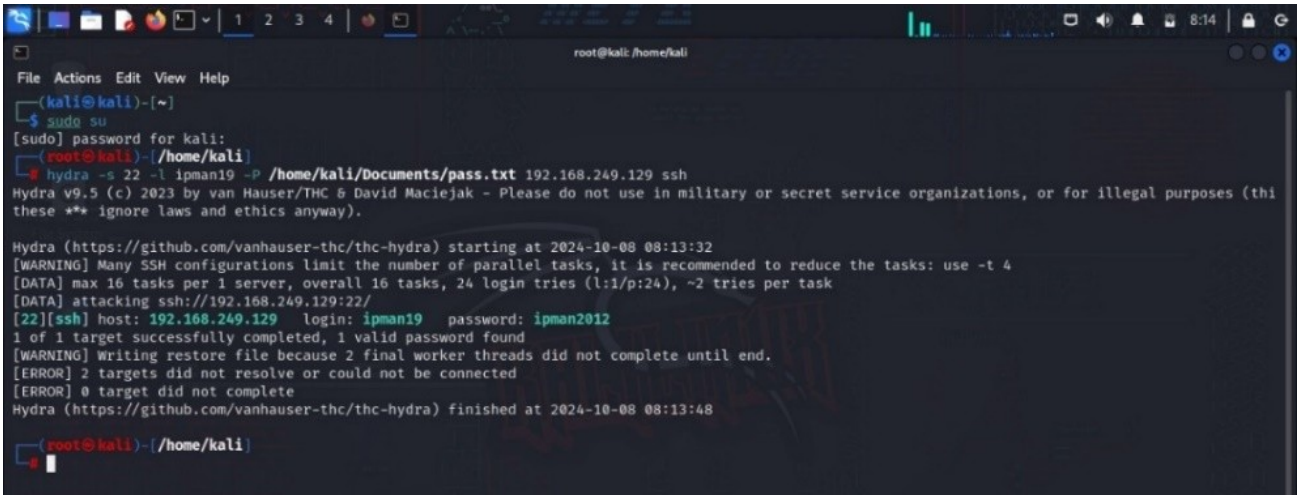

Figure 6: Brute force attack test display before the system is implemented

JOMAse | Received: 25-August-2025 | Accepted: 30-November-2025 | [(69) 3: 264-269]
Published by International Society of Ocean, Mechanical and Aerospace Scientists and Engineers, www.isomase.org., ISSN: 2354-7065 & e-ISSN: 2527-6085

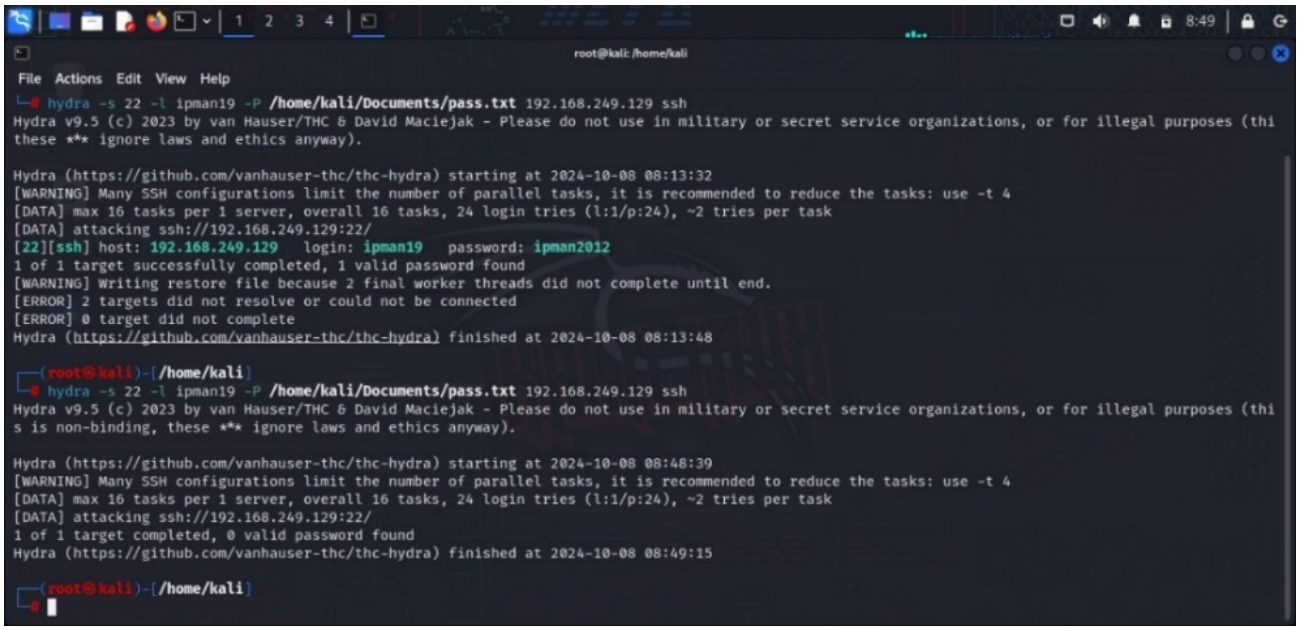

Figure 7: Brute force attack test results after the system is implemented

The system's success in sending notifications also became an important part of this testing. The configured Telegram bot was capable of sending messages containing the latest knock sequences and IP blocking information in real-time to administrators, as shown in Figure 8. This facilitates administrators in monitoring network conditions and responding to threats rapidly.

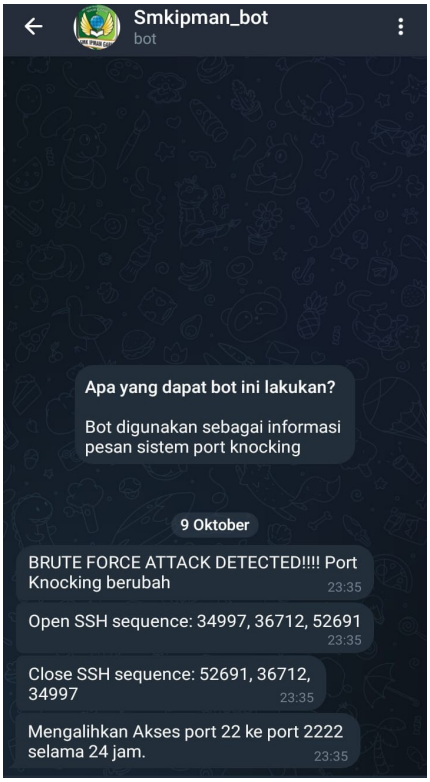

Figure 8: Telegram notification of a network attack

3.3 Test Results Recapitulation

Overall, results from five tests under active system conditions demonstrate that the system is capable of, Concealing port 22 from external port scanning results.

1. Detecting and automatically blocking attacker IPs after three failed login attempts.
2. Automatically randomizing knock sequences.
3. Redirecting attacker connections to the Cowrie honeypot for further activity monitoring.
4. Sending real-time notifications of knock sequence changes and IP status to administrator Telegram.

Test results recapitulation is displayed in Table 2, which contains columns for port scanning results, brute force attack status, notification delivery status, and whether honeypot redirection was successfully performed. From all five attempts, the entire system functioned as designed, demonstrating high effectiveness in securing SSH port access.

Table 2: Data results of all tests

No	Testing Aspect	Condition Before	Condition After
1	Attack Type: Port Scanning	Open	Filtered
2	Attack Type: Brute Force	Password Found	Password Not Found
3	Service Status: Port 22	Open	Filtered
4	SSH Password Attempt	Detected	Not Detected
5	Telegram Notification	–	Sent
6	Traffic Redirection to Honeypot	–	Redirected

3.4 Discussion

Test results indicate that implementing the random port knocking method provides significant impact on network

JOMAse | Received: 25-August-2025 | Accepted: 30-November-2025 | [(69) 3: 264-269]
Published by International Society of Ocean, Mechanical and Aerospace Scientists and Engineers, www.isomase.org., ISSN: 2354-7065 & e-ISSN: 2527-6085

security enhancement. By concealing the SSH port and only opening it for IPs that have performed knocks according to the sequence, the system successfully prevents scanning and illegal login attempts. Compared to conventional systems that only rely on static firewalls, this system is dynamic as it can automatically change knock sequences in response to attacks.

Furthermore, Cowrie honeypot integration provides additional benefits in the form of attacker activity data collection. This information is useful for attack pattern analysis and future network security policy development. The use of real-time notifications through Telegram also enhances monitoring efficiency, enabling administrators to immediately recognize and address potential threats.

Thus, the proposed system proves effective in preventing unauthorized access, strengthening SSH port security, and increasing administrator awareness of suspicious network activities. This approach can be applied in school environments, educational institutions, and small-to-medium organizations requiring simple yet robust network security solutions.

4. CONCLUSION

Based on the design, implementation, and testing results that have been conducted, it can be concluded that the random port knocking-based network security system combined with IPTABLES, Fail2ban, Telegram notifications, and Cowrie honeypot is capable of significantly enhancing SSH port security. This system successfully conceals ports from external scanning, automatically blocks attacker IPs, sends real-time notifications, and redirects suspicious activities to honeypots for further monitoring. As future development, this system can be enhanced with the addition of web-based monitoring panels so administrators can monitor network security status visually and more interactively.

ACKNOWLEDGMENTS

The authors wish to express sincere gratitude to all parties who have provided support, both material and spiritual, enabling this research to be completed. Furthermore, the highest respect and appreciation are extended to all parties involved in research activities at Universitas Garut.

REFERENCES

[1] Iqbal, M., Arini, A., & Suseno, H. B. (2020). Analisa dan simulasi keamanan jaringan Ubuntu Server dengan port knocking, honeypot, iptables, ICMP. *Cyber Security dan Forensik Digital, 3*(1), 27–32. https://doi.org/10.14421/csecurity.2020.3.1.1933.

[2] Wahyu, A. P., Fauziah, K., Nahrowi, A. S., Faiz, M. N., & Muhammad, A. W. (2023). Strengthening network security: Evaluation of intrusion detection and prevention systems tools in networking systems. *International Journal of Advanced Computer Science and Applications*, *14*(9).

[3] Husain, I., Hariyadi, P., Latif, K. A., & Aditya, G. T. (2023). Implementation of port knocking with Telegram notifications to protect against scanner vulnerabilities. *Matrik: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, 23*(1).

[4] Mulyanto, Y., Julkarnain, M., & Afahar, A. J. (2021). Implementasi port knocking untuk keamanan jaringan SMKN 1 Sumbawa Besar. *Jurnal Informatika Teknologi dan Sains, 3*(2), 326–335.

[5] Kizza, J. M. (2017). *Guide to computer network security* (4th ed.). Springer.

[6] Ramadhani, S. T. A., Puri, F. M., & Huda, A. K. (2024). Enhanced security of Linux server-based servers with a combination of iptables and knocking ports. *Jurnal Sistem Telekomunikasi, Elektronika, Sistem Kontrol, Power System & Komputer, 4*(2), 113–124. https://ejournal.uniska-kediri.ac.id/index.php/JTECS/article/view/5541.

[7] Arini, A., Suseno, H. B., & Iqbal, M. (2020). Analisa dan simulasi keamanan jaringan Ubuntu Server dengan port knocking, honeypot, iptables, ICMP. *Jurnal Cyber Security dan Forensic Digital, 3*(1), 27–32. https://doi.org/10.14421/csecurity.2020.3.1.1933.

[8] Dermawan, A., Yuhandri, & Sumijan. (2024). Analisis perbandingan optimalisasi port knocking dan honeypot dengan iptables pada server untuk keamanan jaringan. *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen), 5*(2), 543–556.

[9] Sadiqui, A. (2020). *Computer network security*. Wiley.

[10] Purbo, O. W. (2019). Ubuntu Linux server security analysis and simulation with port knocking & iptable. *International Journal of Basic and Applied Science, 8*(2), 36–46. http://www.ijobas.pelnus.ac.id.

[11] Mardiansyah, A. Z., Abdussyakur, Y. M., & Jatmika, A. H. (2021). Optimasi port knocking dan honeypot menggunakan IPTables sebagai keamanan jaringan pada server. *JTIKA (Jurnal Teknologi Informasi, Komputer, dan Aplikasinya), 3*(2), 189–199. https://jtika.if.unram.ac.id/index.php/JTIKA/article/view/144.

[12] Ernawati, T., Kholid, I., Dahlan, & Rohmayani, D. (2024). Case study in network security system using random port knocking method on the principles of availability, confidentiality and integrity. *Jurnal Online Informatika, 9*(1), 41–51. https://doi.org/10.15575/join.v9i1.1254.

[13] Suharto, N., Jeinever, I. R. P., & Rasyid, A. (2018). Penerapan sistem keamanan jaringan menggunakan random port knocking berbasis Raspberry Pi yang dikirim melewati Telegram. *Jurnal Jaringan Telekomunikasi, 7*(2), 2407–0807.

[14] Marzuki, I. (2017). Perancangan dan implementasi sistem keamanan jaringan komputer menggunakan metode port knocking pada sistem operasi Linux. *Jurnal Teknologi Informasi Indonesia, 2*(2), 18–24.