Journal of Engineering and Sustainable Development
Vol. 29, No. 05, November 2025
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.2809

Enhancing Server Security: Implementation and Evaluation of the Port Knocking Method on Ubuntu Virtual Servers

Sadeer Sadeq[1*], Wid Badee Abdulaziz[2], Rafid Najim Abdullah Alsaadi[3], Mabrouka M.A. Algherini[4]

[1]Scientific Affairs Department, University of Information Technology and Communications, Baghdad, Iraq
[2]Electronic Computer Center, University of Information Technology and Communications, Baghdad, Iraq
[3]Construction and Projects Department, University of Information Technology and Communications, Baghdad, Iraq
[4]Department of Science and Computer Engineering, Higher Institute of Science and Technology, Sokna, Aljufra, Libya

[*]Email: sadeer.abduljabbar@uoitc.edu.iq

Article Info

Received	**25/06/2024**
Revised	**05/09/2025**
Accepted	**15/09/2025**

Abstract

Server security is the most important thing for administrators. The server has service access via port 22, which is the most crucial point to secure because illegal access to the system is common. To overcome this problem, a firewall is implemented to act as a barrier. However, firewall users themselves are still less effective because of how it works, which closes all access without caring about anyone connected to the network. To address these issues, the port-knocking method is used in server security. The technique works by opening and closing Port 22 blocks using a firewall that detects knock attempts on network devices. Tests were conducted using port scanning, a DDoS attack, and brute-force attacks to evaluate the port-knocking method's performance in securing the server. With many tests —30, 15, and 26 times—success percentages of 0%, 90%, and 100%, the port knocking method can secure the server against unauthorized access and overcome problems caused by firewalls.

Keywords: Brute force, Client, Firewall, Port knocking, Port 22, Server

1. Introduction

A computer network usually consists of a server and a client. An administrator controls the server. One way is by running a remote server. The administrator who remotes a server must be the person authorized to access it [1]. However, some attackers deliberately gain access to the system, make changes, and damage the server. One way to improve a server's security is to use a firewall. But currently, it still has weaknesses. Firewalls cannot distinguish between trustworthy users. Firewalls can only distinguish IP addresses that they assume are used by untrusted people. So, solutions are sought to reduce existing weaknesses [2]. One of these methods is port-knocking. A security method that can close gaps and overcome firewall-imposed access restrictions is Port knocking. Port knocking is a simple method that provides remote access without leaving port 22 open, thereby protecting the server from port scanning, DDoS attacks, and brute-force attacks.[3] Port knocking works by closing the Port; only specific users can access it by knocking on the targeted Port first. Meanwhile, the firewall closes all ports without knowing which users have access rights [4]. Many organisations need more IT resources to implement remote strategies. After deploying workers and students outside the perimeter, managing device sprawl, patching, and securing hundreds of thousands of endpoints becomes more difficult. Hub-and-spoke architectures have connected remote users and branch offices to the corporate network for decades. Now that more employees work remotely and more applications run in the cloud, this antiquated approach complicates business operations. Regulations make remote work difficult by requiring the use of unapproved systems, devices, or people. The challenge is to secure users' access to those systems using VPNs or other network solutions. Researchers use the Port Knocking method to overcome attacks on ports in a computer network, a security system specifically designed for networks. Port knocking works by closing all existing ports; only specific users can access a specified port after knocking first. Therefore, a method that meets these conditions is needed to ensure security and reduce attacks on the server, allowing trusted users to access it even when the ports appear closed. One security method that meets these standards is Port Knocking.

Server security is paramount for administrators. The server has service access via port 22, which is the critical point that must

be secured due to frequent unauthorised access to the system. To address this issue, a firewall is implemented as a barrier; however, its effectiveness is diminished by its operational method, which indiscriminately restricts all access, regardless of the individuals connected to the network. The technique operates by opening and closing Port 22 blocks via a firewall that identifies knock attempts on network devices.

1.1. Network Security

Network security is a preventive measure against attackers who attempt to access a computer network through unauthorized means. A system is used to protect the network from various external threats that can damage it [5]. We'll limit internet access to our SSH daemon, mainly via our eth0 interface. Our firewall will be temporarily changed to allow SSH connections from our PC once a predetermined order of ports has been accessed. This traffic will be permitted since a web server is assumed to use port 22 by default. After establishing our essential allowed connections, we can use our KNOCKING chain to carry out the knocking logic by rerouting any traffic not covered by the previously mentioned rules.

1.2. Servers

A server is a computer system that provides exceptional services. The data stored on the server is complex documents and information. The service aims to meet clients' needs and provide users with access to information. Servers have an essential role in sending and receiving information more quickly. A server is a large-scale computer network that accommodates components such as processors and large-capacity RAM [6].

1.3. SSH (Secure Shell)

Secure shell protocol in a computer network that functions to help users exchange data, transfer data, and operate remote computers safely. SSH provides greater security than Telnet and FTP. SSH provides security in computer networks, especially against Man-in-the-Middle (MITM) attacks. SSH uses port 22 to connect to a computer network. [7].

1.4. Port knocking

Port knocking is a critical concept for securing the services the server provides. Used to open access to specific ports that have been closed by the firewall by sending certain packets or connections in the form of TCP, UDP, or ICMP protocols. The host initiates the connection according to the knocking rules applied, so the firewall will automatically grant access to the blocked Port. Port knocking is a new technology that promises to secure remote services better. This technology can close all ports, including TCP ports, until the user authenticates using a port-knock sequence. All ports remain closed, making the server invisible to port scanners [8]. How port knocking works, which functions to secure by tapping first to be able to enter the secured server via port 22 (SSH), is as follows:

1.4.1 Deflection of port 22

Fig. 1 shows that the client cannot access port 22 because the firewall blocks it. Meanwhile, to access the server via port 22, the client must enter the knock format.

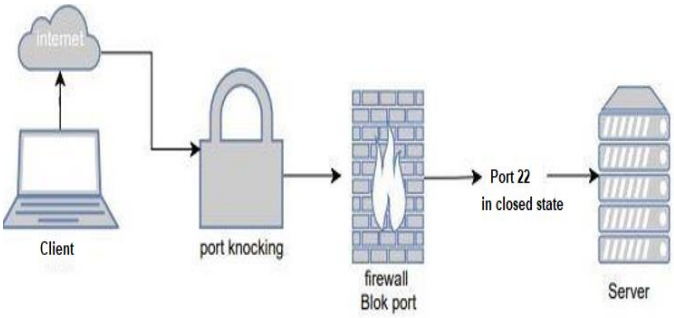

Figure 1. Port block 22.

1.4.2 Knocking

In Fig. 2, the client process pings the server by simultaneously entering the knock sequence 70001, 80002, and 90003 over 3 seconds.

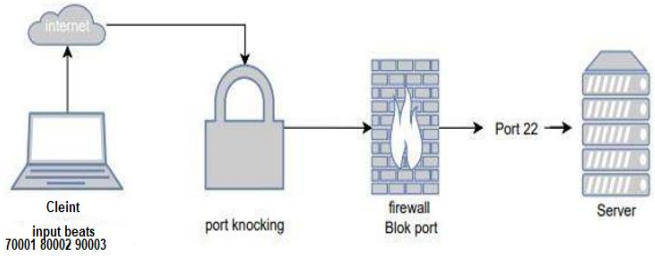

Figure 2. Knocking process.

1.4.3 Beat output.

In Fig. 3, the firewall output process blocks connections and interprets (deciphers and reads) passwords, resulting in authentic Port knocking. The firewall performs specific tasks based on the contents of the knocking Port, allowing and monitoring traffic based on the user's IP, and recording it in logs.

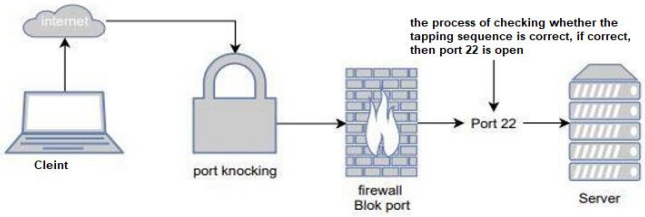

Figure 3. Output process.

1.4.4 Open port process.

In Fig. 4, the process opens a remote port 22 on the server.

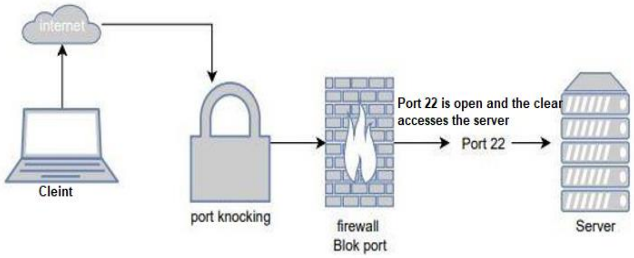

Figure 4. Port 22 is open.

1.4.5 Port 22 is closed.

In Fig. 5, the client process closes the Port by entering 90003 80002 70001, so that all activities and the status of port 22 are closed.

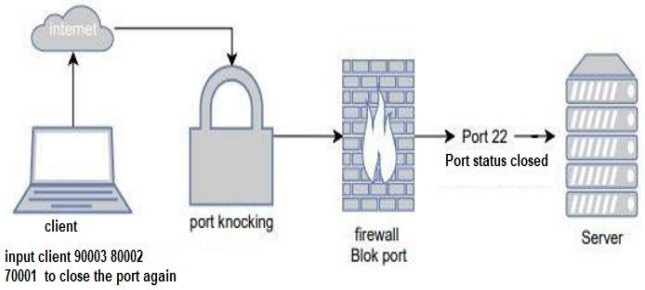

Figure 5. Port 22 Closed.

Ports a+1000, b+1500, and c+2000 are destination ports for sending data packets and serve as knock ports. Destination ports to be opened and closed as shown in Table 1. So, the knock the user uses to open port 22 is 1000, 1500, and 2000. Meanwhile, to close the Port, use 2000, 1500, 1000. After completing the knock format, there is also a port-knocking mechanism. Krzywinski said the working pattern of the port-knocking method has the following stages:

Table 1. Determination of Knock Format.

Port Knock Number	Destination Port
a+1000, b + 1500, c + 2000	*Abc*

1. The client connects to one of the server's ports, but the firewall blocks the connection.
2. The client connects to the port sequence defined in the port daemon configuration file to the server computer by sending a SYN packet; during this phase, the client does not receive a response.
3. The Daemon records the experiments that the client has carried out.
4. The client disconnects with the Port and returns the SYN packet. The Daemon rewrites the firewall so it cannot reconnect to port 22 [9].

1.5. Virtual Private Servers (VPS)

A virtual private server is a virtualization technology. A physical server is divided into several VPS so that it works like an independent server. VPS has Full Root access, including the operating system and settings for scripts, user accounts, the file system, and server resources such as CPU and RAM. VPS allows multiple operating systems to run simultaneously on a single physical server machine. This can be done without repartitioning and rebooting. The VPS provided will run the desired operating system. Users can create an operating system (Linux) as the operating system used on the server (host) and run other operating systems [9].

1.6. Firewalls

A firewall is a vital network component for controlling network traffic. It aims to enable controlled connectivity between clients and servers, preventing unauthorised communication within the computer network. Hardware firewalls are used to limit network access. The access control mechanisms offered by firewalls depend on a set of rules that administrators define and then apply to every packet that flows through the firewall [10].

1.7. Putty

Putty is an SSH and Telnet client developed by Simon Tatham for the open-source Windows platform that implements the SSH, Telnet, and Rlogin network protocols. The protocol can run remote sessions on a computer over a long-distance network [11].

1.8. Port Scanning

Port scanning is a hacker's activity that determines whether a target host is active on a network. The scanning results include IP addresses, operating systems, services, and running applications. This information helps determine the method to attack the system [12].

1.9. DDOS Attacks

A DDoS attack targets a victim by exploiting a weakness in the victim's system. The primary targets of DDoS attacks are resources such as bandwidth, CPU, and network capacity [13].

1.10. Brute force

A brute-force attack is a method of breaking a secret code by trying all possible keys. Crackers use this method to obtain accounts illegally. Apart from the time required to look, the time needed to find the correct password cannot be determined [14].

1.11. Unified modelling language (UML)

The Unified Modeling Language is a collection of modelling conventions used to describe a software system in terms of objects. UML is used as a design tool and is often adopted by users because it is not affected by software, hardware, operating systems, networks, databases, or programming languages [15].

1.12. Website-Based Applications

1.12.1. Hypertext markup language (HTML)

Hypertext Markup Language (HTML) is a language used to write web pages. Designed without depending on a specific platform. The main characteristics of HTML documents are the presence of tags and elements [16], [17].

1.13 Objective of the Study

Given the advantages of Port knocking over firewalls, the research will implement security measures on virtual VPS servers and use a web-based system monitoring system.

2 Literature Review

To support this research, the author draws on literature related to the study's title and topic. The research was carried out under

the title "Network Security Applications Using the Port Knocking Method". The results show that the port-knocking method can determine which ports the client can access or not [18]. The subsequent related research focuses on implementing port knocking, which can overcome authentication problems and brute-force attacks [19]. Other research results show that the port-knocking method successfully secures the router from unauthorized access [20]. Given the advantages of Port knocking over firewalls, the research will implement security measures on virtual VPS servers and use a web-based system monitoring system. Based on research conducted by Christian [21], they designed the configuration of user access rights on the Port using Microtek Router. They tested security using the Port Knocking method and the Tarpit Firewall action. The Port knocking method is a technique that temporarily closes access to the Port. Port knocking on this Microtek Router can be accessed via Winbox and the web service provided by MikroTik in Winbox Webfig. However, there are other methods besides accessing the Winbox Port: SSH and HTTP Ports, since Winbox can also be accessed via SSH and via web services. Marzuki's [22] research primarily focuses on network security systems against attacks that can disrupt or damage networks. The case study involves an SSH network running on Port 22 because SSH services are generally the primary targets of attackers. The security method used in this study is Port knocking. Based on the research results, the designed system can enhance network authentication security by keeping the Port closed to the public. The port-knocking method is effective at increasing network security because clients must provide a "knock" to a specific port to access the SSH service.

3. Research Methods

The research method outlines the implementation steps to make it easier for researchers to conduct research in a structured manner. The study's structure is shown in Fig. 6.

3.1. Data Collection Methods

The data collection method involves collecting data to support research through observation. The observation method is a data collection technique that uses direct observation; for example, in this research, observation is through client log records, enabling the researcher to obtain data from them.

3.2. Needs Analysis

Process analysis needs information about the software and hardware relevant to the research. This research analyses needs by looking for what will be needed in the system design. For example, the testing section requires a port-scanning tool or application to test the Port that targets attacks on the server.

3.3. System Design

System design is the process from start to finish that results in a system. This system will be used as a research object to conduct defense and tests to obtain data.

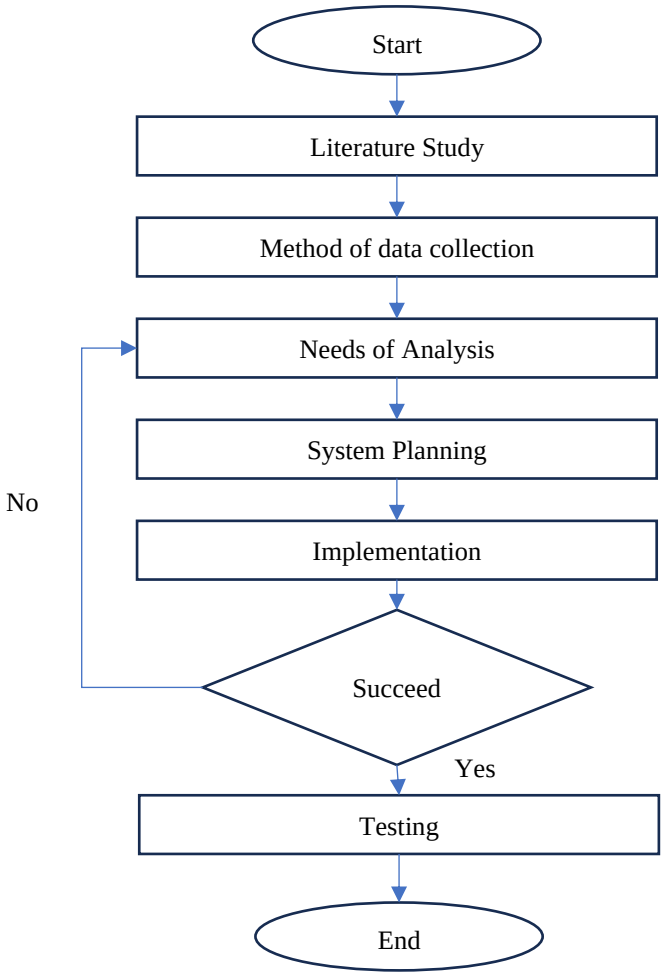

Figure 6. Research Structure Diagram.

3.4. Implementation

Implementation is the stage of fulfilling all requirements to build a system. Implementation starts by installing iptables, which filter network traffic; blocking the SSH port as the server's primary service; installing knock; connecting the server to the database; and displaying activity from the intruder's IP in the Monitoring application.

3.5. Testing

Testing is a stage carried out by researchers in which they knock the system. The system records the "knock" activity and uses it as data for research on the success or failure of the "knock" within the system. Testing was also conducted using several attacks, including port scanning, DDoS, and brute-force. The port-scanning test is performed by scanning the target IP address.

4. Proposed Design

A system description is an overall explanation of the system. The system will generally be built to test a port-knocking method for server security.

4.1. System Architecture Design

System design is carried out to determine the flow of system creation in more detail. The research uses Ubuntu 16.04 LTS (HVM) as the server operating system; the client uses Windows, which is simulated as an attack. The design is carried out as in Fig. 7.

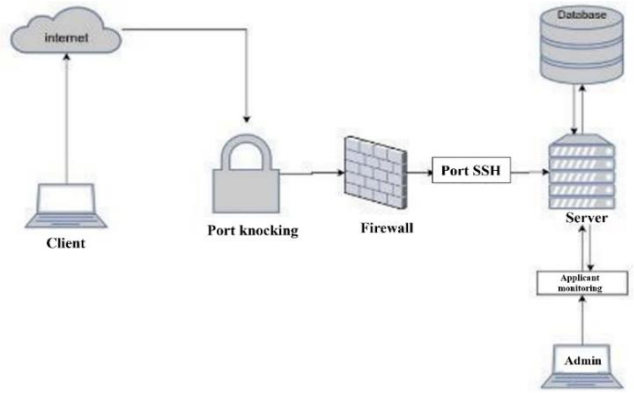

Figure 7. Design of the entire system.

Fig.7. depicts the step-by-step flow design with a server as the primary research topic.

4.2. Software Design

Software design is carried out for system requirements. Apart from using hardware architectural design, software sessions also carry out the design. In software design, it connects applications, servers, and databases. The application functions as a place to monitor activity from clients who have "knocked" the system. The server functions as a system protected against unauthorized access by intruders or by users without the admin's permission or knowledge. The database is a storage place for log data generated by system access.

The following is an explanation of the flow chart in Fig. 8

1. Before knocking, the client gathers information about the target system by scanning it with Nmap. Intending to get information, including whether port 22 is open or closed. If the Port is open, the client can access the system directly.
2. Port 22 is closed; the client can access the system by entering the installed knock rules. If it is appropriate, port 22 is opened, and the client can access the system. If it is not suitable, the client must knock again to enter.
3. Client logs are stored in the database and displayed in the Monitoring application.

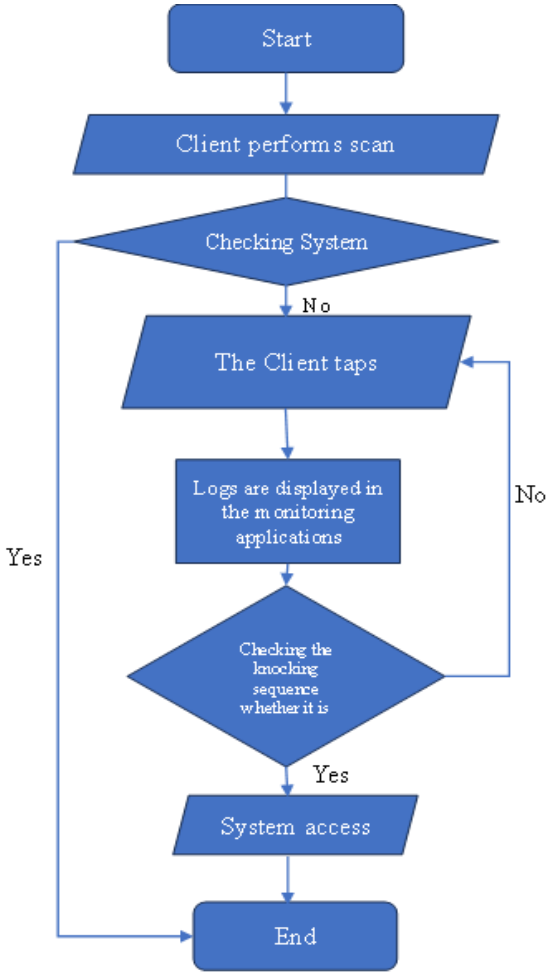

Figure 8. Software Designer.

4.3. Database Design

The database is a storage place for research-supporting data that will be included in the system. Database research using MYSQL. The following is the database design for the system.

1. Administrator Login as Table 2

Table 2. Admin Login.

S.No.	Name	Type	long	Existence
1	*Id_admin*	TINYINT	2	Primary
2	*username*	VARCHAR	15	
3	*password*	VARCHAR	15	
4	*Name_Admin*	VARCHAR	30	

2. Activity as Table 3.

Table 3. Activities.

S.No	Name	Type	long	Existence
1	activity_ID	*int*	3	Auto_increment
2	IP	VARCHAR	30	
3	data	Date Time	0	
4	SSH status	TINYINT	2	
5	Knock status	VARCHAR	0	
6	Time out	Data time	0	

4.4. System Test Design

The system test design aims to ensure that every test runs successfully. The test design included $port\ scanning, DDoS\ attacks$, and $brute-force$ attacks.

4.4.1. Port scanning attack plan

$Port-scanning\ attack$ on the LANattack LAN to gather information about the target server. The attack process will be carried out by scanning ports using the $nmap$ application installed on the computer, which will act as a client and serve as the attacker. Performing client testing runs the following command.

"$Nmap$ 149.28.137.100"

The command above is a script run to scan the target IP.

4.4.2. DDOS attack test design

DDoS attack testing is carried out by sending large numbers of packets. Testing DDoS attacks on servers causes data traffic to become congested. Testing was carried out by sending a ping. $94.237.73.226\ -t\ -l\ 6500$. Sending pings to the target IP continuously [23].

4.4.3. Brute force test design

Brute-force testing is designed to understand better the steps involved in implementing the test. Brute-force testing is conducted to find SSH keys embedded in the system, enabling attackers to access it after successfully obtaining the key [24].

5. Test Results and Discussion

The main goal of this research is to develop solutions to problems and to design network security systems. This experiment involves three steps: network security design, daemon configuration using port knocking, and server computer by sending a SYN packet, and initial and final testing.

5.1. Port Knocking Implementation

5.1.1. IP tables

Iptables aims to make it easier for the system to process data on packages connected to the system.

Fig. 9 shows the scripts that function in this research, such as -A helps add rules to tables on the firewall, INPUT is used to manage incoming connections, -m helps manage incoming traffic, --ctstate ESTABLISHED, RELATED is used to determine connections, and -j ACCEPT is used to filter traffic.

```
[options]
        UseSyslog
        logfile = /var/log/knockd.log

[openSSH]
        sequence    = 70001,80002,90003
        seq_timeout = 5
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 90003,80002,70001
        seq_timeout = 5
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

Figure 9. Configuration of the iptables − persistent package.

In Fig. 10, you can see the command to block port 22 using -p tcp, which targets TCP traffic, and -D port 22 to mark it as a blocked port.

```
root@rosa:~# iptables -A INPUT -p tcp --dport 22 -j REJECT
```

Figure 10. Block Port 22 Command.

Fig. 11 shows port 22 in a closed state.

```
Starting Nmap 7.01 ( https://nmap.org) at 2023-11-20 04:22 UTC
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try
using .. system-dns or specify valid servers with-dns-servers
Nmap scan report for 94.237.73.226
Host is up (0.0000040s latency).
Not shown: 998 closed ports
PORT    STATE    SERVICE
22/tcp  filtered  ssh
80/tcp  open      http

Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
```

Figure 11. Port 22 is closed.

5.1.2. Port Knocking

At this stage, install Port Knocking on the server. $Knock$ on the Linux operating system.

Fig. 12 is a script for installation. Knockd, using apt-get, which functions to install, delete, and update on the Linux OS

```
root@skripsi-rosa:~# sudo apt-get install knock
```

Figure 12. Install knock.

In Fig. 13, you can see the program run to access the knock activity storage directory on the system.

```
root@skripsi-rosa:~# nano /var/log/knockd.log
```

Figure 13. Log file.

File: There are three main parts to the file $knockd.Conf$ as Fig. 14. The first section is the options section, which shows the log file's location so that all knock activity can be seen. The second part configures the destination port, and the third part closes it again.

```
root@rosa:~# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACC
EPT
```

Figure 14. $Knockd.conf$.

In Fig. 15, 1 indicates it is starting, and 0 indicates a sudden knock that neither starts nor stops. Fig. 16 explains the active knock status.

```
START_KNOCKD=1

# command line options
KNOCKD_OPTS="-i eth0"
```

Figure 15. Start to knock.

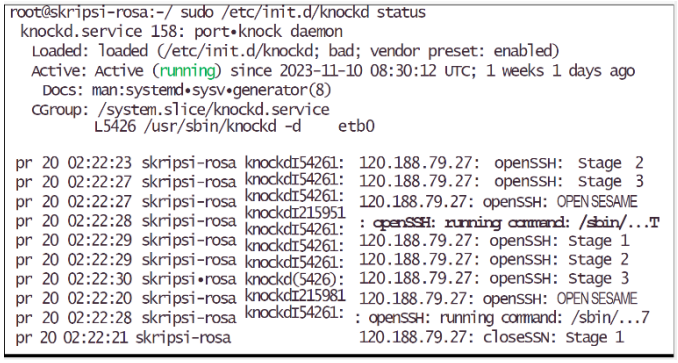

Figure 16. Active knock status

5.2. Implementation of Monitoring Applications

5.2.1. Database implementation

Database implementation aims to assess the success of the previously created database design.

User Table: User table to store data from users who access the monitoring application.

In Fig. 17, the data is stored in the user database—data of a user who can log in to access the monitoring application. To access the monitoring application, you must use data stored in the database.

Log table: The log table stores information about intruders who "knock" the server, as shown in Fig. 18.

5.2.2. Implementation of the Monitoring Application

Implementing the monitoring application enables monitoring intruder activity on the server. The monitoring application is implemented to assess the success of the previous design.

In Fig. 19, you can see the initial display; in the top-right corner, you can see the words "login" to access the monitoring application.

In Fig. 20, you can see the display entering the username and password as a requirement for the user to enter the application.

The dashboard displays several components, including an activity information table, a graph of the knock status activity, and a count of clients scanning the IP server.

In Fig. 21, you can see the action table for the attacker, shown in Fig. 16. The table shows the intruder's source, the "knock" time sequence, and whether they succeeded in accessing the system via the SSH port. The data recorded in the *knockd* folder shows IP activity that has carried out multiple knock attempts on the system.

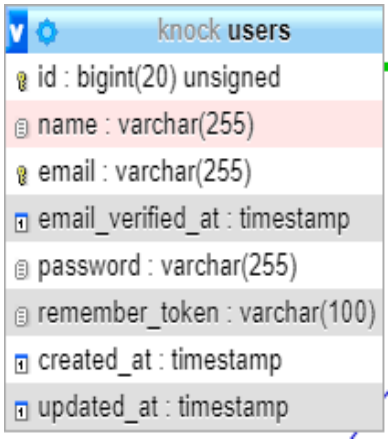

Figure 17. User table.

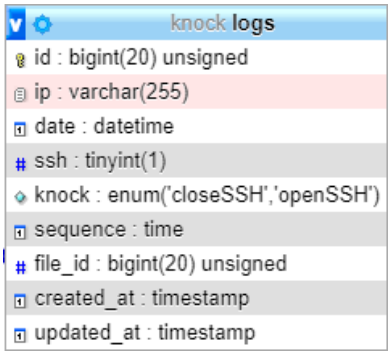

Figure 18. Log Table.

Figure 19. Application front page.

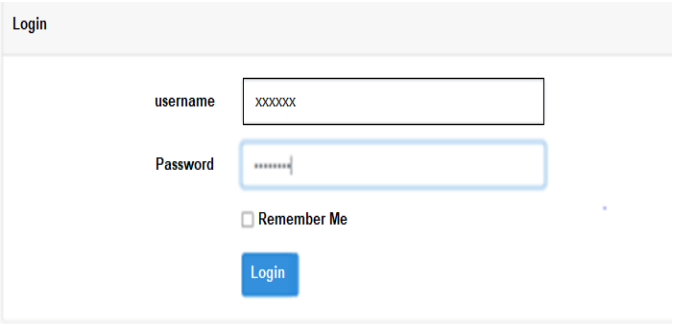

Figure 20. Login Page Display.

5.3. Effectiveness of the Port Knocking System

5.3.1. Port Knocking Test

Knock testing is performed to verify that Port knocking on the server is implemented successfully.

In Fig. 22, you can see the success of the client access by inputting. 90003 80002 7000, The client successfully accessed the server that had implemented the port-knocking method.

No ↑↓	IP ↑↓	Date/Time ↑↓	Status knock ↑↓	Sequence Timeout ↑↓
25	45.129.33.49	2023-11.22 00.25.31	openSSH	01.40.00
27	193.26.62.63	2023-11.23 00.08.19	openSSH	06.41.00
29	192.241.239.32	2023-11.23 00.11.09	openSSH	10.30.00
32	78.262.321.20	2023-11.23 00.12.27	openSSH	11.22.00
35	94.102.32.62	2023-11.23 00.13.10	closeSSH	12.39.00
41	192.240.361.12	2023-11.23 00.14.39	closeSSH	14.19.00
45	196.54.124.221	2023-11.23 00.14.58	openSSH	16.21.00
50	185.261.30.42	2023-11.23 00.15.09	closeSSH	18:02.00
53	193.22.202.218	2023-11.23 00.16.11	openSSH	19.11.00
57	167.25.73.32	2023-11.23 00.17.37	closeSSH	20.22.00

Figure 21. Attack Table.

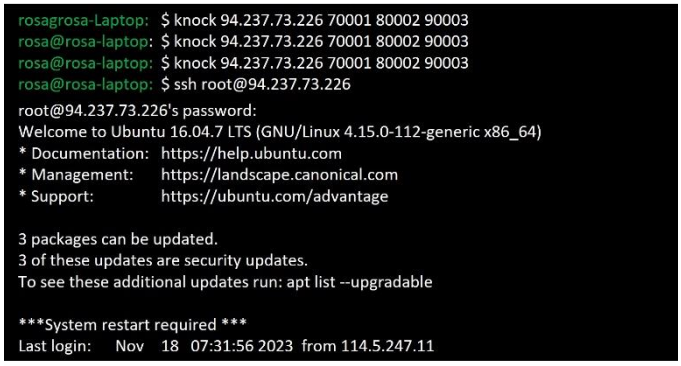

Figure 22. Access Success.

As shown in Fig. 23, the client connects to the server in the wrong order, so the client does not connect to the host IP address and $port$ 22 (SSH port) appears to be closed.

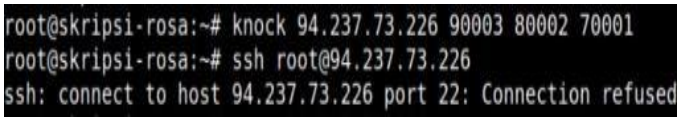

Figure 23. Unsuccessful in accessing the server.

5.3.2. Port Scanning Testing

Port scanning is used to gather information about the server, such as whether a destination port is open or closed. At the port scanning stage, we use the NMAP (Network Mapper) tool on the server to scan the IP address to check the status of port 22. Testing is carried out during Port knocking before and after implementation on the server.

As shown in Fig. 24, port 22 on the server is open. This makes it easier for intruders to gain access to the server.

Figure 24. open Port.

In Fig. 25, it can be seen that. Port 22, a focal point for research in a closed state, makes it challenging for intruders to access the server.

Table 4 shows the results of a port-scanning attack aimed at finding information on port 22. After 15 tests, the results indicate that port 22 is closed.

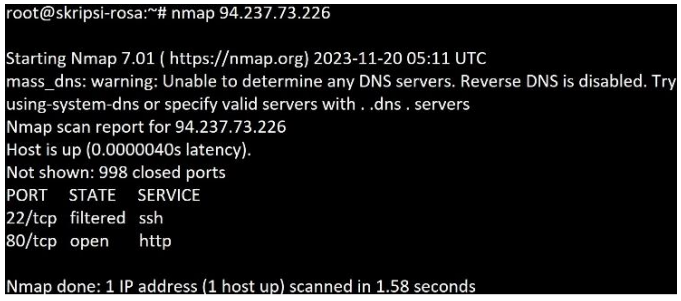

Figure 25. Closed Port.

In Fig. 26, the results of the DDoS attack test are shown. Testing is done by pinging the server's IP address from CMD on Windows.

5.3.3. DDoS Attack Testing

DDoS attack testing will be conducted to verify the server's security after implementing the port-knocking method.

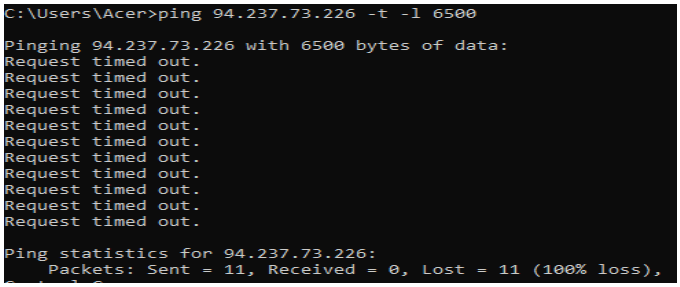

Figure 26. DDOS Attack.

Table 5 presents results from DDOS attack testing on servers that have implemented the port-knocking method. This data shows that using Port knocking minimizes the server's exposure to DDoS attacks.

5.3.4. Brute force testing

Brute-force testing was conducted in this research to identify the key to port 22 on the server. Testing was carried out using the Hydra tool. The system is tested by inputting one of the scripts (Fig. 27).

The brute-force test is unsuccessful; the information shown in Fig. 27 indicates that it cannot connect to port 22.

From Table 6, you can see the brute-force test results on the system. The results show that access to the system is closed. The attack to find the password on port 22 was unsuccessful because of the port-knocking method implemented on the server.

Table 4. Port Knocking.

				Port 22 Scan results	
S.No	**Source IP**	**Destination IP**	**Time**	**Closed**	**Open**
1	114.5.11.162	149.28.137.100	3.48.00	√	
2	209.17.96.154	149.28.137.100	3.49.00	√	
3	80.82.65.74	149.28.137.100	7.56.00	√	
4	193.93.62.61	149.28.137.100	8.03.00	√	
5	124.156.240.79	149.28.137.100	9.19.00	√	
6	88.202.190.139	149.28.137.100	10.33.00	√	
7	193.93.62.61	149.28.137.100	12.09.00	√	
8	129.227.129.18	149.28.137.100	13.33.00	√	
9	45.129.33.39	149.28.137.100	13.52.00	√	
10	83.97.20.35	149.28.137.100	14.13.00	√	
11	45.129.33.59	149.28.137.100	15.18.00	√	
12	193.93.62.61	149.28.137.100	16.20.00	√	
13	192.241.239.94	149.28.137.100	17.40.00	√	
14	78.128.113.42	149.28.137.100	18.13.00	√	
15	94.102.51.95	149.28.137.100	19.20.00	√	

Table 5. DDos Attacks.

			Server Status using the port-knocking method	
S.No	**Source IP number**	**Destination IP number**	**Succeed**	**Failure**
1	10.45.2.106	94.237.73.226	√	
2	10.100.13.8	94.237.73.226	√	
3	10.45.10.64	94.237.73.226	√	
4	10.45.10.584	94.237.73.226	√	
5	10.45.10.203	94.237.73.226	√	
6	114.5.245.162	94.237.73.226	√	
7	80.82.65.17	94.237.73.226	√	
8	45.163.62.14	94.237.73.226	√	
9	83.62.20.59	94.237.73.226	√	
10	209.17.96.154	94.237.73.226	√	
11	193.63.256.31	94.237.73.226	√	
12	45.36.21.52.3	94.237.73.226	√	
13	94.62.354.12	94.237.73.226	√	
14	88.202.190.39	94.237.73.226	√	
15	192.168.32.33	94.237.73.226	√	

16	192.168.100.33	94.237.73.226		√
17	192.168.100.11	94.237.73.226		√
18	10.45.5.129	94.237.73.226	√	
19	10.45.10.65	94.237.73.226	√	
20	10.45.5.37	94.237.73.226	√	
21	10.45.6.112	94.237.73.226	√	
22	10.45.6.36	94.237.73.226	√	
23	10.45.10.203	94.237.73.226	√	
24	10.45.5.74	94.237.73.226	√	
25	10.45.5.37	94.237.73.226	√	
26	10.45.5.92	94.237.73.226	√	
27	10.45.5.129	94.237.73.226	√	
28	10.45.5.29	94.237.73.226	√	

Table 6. Brute Force Attacks

S.No.	Brute Force Attacks	Objective	Date	Time	Status
1	Testing 1	Port 22	19-11-2023	5.26	Access Denied
2	Testing 2	Port 22	19-11-2023	6.25	Access Denied
3	Testing 3	Port 22	19-11-2023	7.28	Access Denied
4	Testing 4	Port 22	19-11-2023	8.36	Access Denied
5	Testing 5	Port 22	19-11-2023	9.31	Access Denied
6	Testing 6	Port 22	19-11-2023	9.26	Access Denied
7	Testing 7	Port 22	19-11-2023	10.25	Access Denied
8	Testing 8	Port 22	19-11-2023	11.28	Access Denied
9	Testing 9	Port 22	19-11-2023	12.36	Access Denied
10	Testing 10	Port 22	19-11-2023	13.26	Access Denied
11	Testing 11	Port 22	19-11-2023	14.38	Access Denied
12	Testing 12	Port 22	19-11-2023	15.25	Access Denied
13	Testing 13	Port 22	19-11-2023	16.28	Access Denied
14	Testing 14	Port 22	20-11-2023	8.36	Access Denied
15	Testing 15	Port 22	20-11-2023	9.46	Access Denied
16	Testing 16	Port 22	20-11-2023	10.26	Access Denied
17	Testing 17	Port 22	20-11-2023	11.25	Access Denied
18	Testing 18	Port 22	20-11-2023	12.28	Access Denied
19	Testing 19	Port 22	20-11-2023	13.36	Access Denied
20	Testing 20	Port 22	20-11-2023	14.39	Access Denied
21	Testing 21	Port 22	20-11-2023	15.26	Access Denied
22	Testing 22	Port 22	20-11-2023	17.25	Access Denied
23	Testing 23	Port 22	20-11-2023	19.28	Access Denied
24	Testing 24	Port 22	20-11-2023	20.36	Access Denied
25	Testing 25	Port 22	20-11-2023	21.79	Access Denied
26	Testing 26	Port 22	20-11-2023	22.26	Access Denied

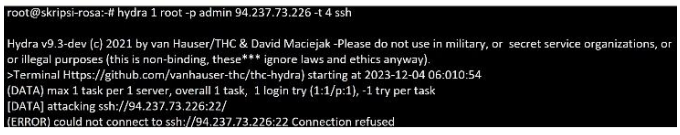

Figure 27. Brute Force Testing.

5.4. Discussion

Based on the literature survey [24]-[27], very few works are available on the security of Ubuntu Virtual Servers. Most works are based on Windows and Linux operating systems [18]. However, the current research applies to the Ubuntu operating system. Idhom et al. [18] worked on network Security Applications Using the Port Knocking Method with Telnet, SSH, port 80, Nmap, PuTTY, Chrome, and Hydra software. The research aims to secure data on the server from unauthorized intruders. Based on the research conducted by Mulyanto et al. [24], improving network security can help administrators secure Microtek Router boards on computer networks in six stages. The current research follows a three-step process, achieving high accuracy in network security results, making it superior to previous works [18], [24]. The success rate of the current work using port 22 is 100%, compared with other works [24]-[26]. This research case centers on a firewall deficiency that needs to be more flexible to block Port 22. When it is

necessary to communicate between client servers in a computer network, the firewall blocks it because it believes the client is not in a permitted location, even though the communication is essential. Necessary for smooth work. During implementation, Port knocking is supported by the Ubuntu LTS 16.04 operating system (OS), which also serves as the server OS. In this research, the virtual private server (VPS) is used as a system that will be secured using the port-knocking method.

To see the defence against the Port Knocking method for securing servers from intruders, several attacks include knocking the system, port scanning, DDoS attacks, and brute-force attacks. Knock testing on the system is performed to assess the success of implementing the method, and the results indicate that the process has been successfully implemented. This can be proven by accessing the server, which must be knocked first. Otherwise, the host port 22 is in a closed status. Port scanning is used to gather information about the target by scanning its IP address. The results of the tests showed that port 22 was closed; across 15 tests, it was 100% closed. This test can also verify the successful implementation of the port-knocking method on the server.

6. Conclusions

Based on research results, the port-knocking method secured the server with an open port gap network service by blocking port 22 and knocking the server to open the SSH port (22) whenever the service was needed. During the port scanning attack, the SSH port (22) was found to be closed, and port knocking was implemented on the system. Based on research results, Port knocking has successfully overcome the problem of port blocking caused by firewalls. Based on test results, port scanning, DDoS attacks, and brute-force port-knocking methods can be used to secure the server.

Based on the test results, suggestions for further research include adding other features to the system to increase security. Because methods that rely solely on knocks have weaknesses, the server cannot be accessed when the port-knocking Daemon dies.

Acknowledgements: The authors are thankful to the reviewers and our co-staff members for their valuable comments on improving the article**.**

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Author Contribution Statement

Sadeer Sadeq determined the objective, problem statement, and proposed work.

Wid Badee Abdulaziz was responsible for data collection and determining the objective.

Rafid Najim Abdullah Alsaadi proposed the work.

Mabrouka M.A. Algherini participated in data collection and stated the problem

References

[1] A. Sallow, H. Dino, Z. Ageed, M. Mahmood, and M. Abdulrazaq, “Client/Server Remote Control Administration System: Design and Implementation”, *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, vol. 3, no. 2, pp. 5–11, 2020, Available: https://ijmrap.com/wp-content/uploads/2020/07/IJMRAP-V3N1P95Y20.pdf

[2] A. U. Nabi, M. Ahmed, and A. Abro, “An Overview of Firewall Types, Technologies, and Functionalities,” *International Journal of Computing and Related Technologies*, vol. 3, no. 1, pp. 10–16, Aug. 2022, Accessed: Jun. 08, 2024. [Online]. Available: http://ijcrt.smiu.edu.pk/ijcrt/index.php/smiu/article/view/126/36

[3] M. Nur *et al.*, “The Effectiveness of the Port Knocking Method in Computer Security,” *International Journal of Integrative Sciences*, vol. 2, no. 6, pp. 861–868, Jun. 2023, https://doi.org/10.55927/ijis.v2i6.4526.

[4] H. Mursyidah, *et al.*, “Analysis and implementation of the Port Knocking method using Firewall-based Mikrotik RouterOS,” *IOP Conference Series: Materials Science and Engineering*, vol. 536, no. 1, p. 012129, Jun. 2019, doi: https://doi.org/10.1088/1757-899x/536/1/012129.

[5] J. Junquera-Sánchez, C. Cilleruelo, L. de-Marcos, and José-Javier Martinez-Herréiz, "C-Lock: Local Network Resilient Port Knocking System Based on TOTP," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–9, Jan. 2022, doi: https://doi.org/10.1155/2022/9153868

[6] R. Sinha, “A Study On Client Server System In Organizational Expectations,” *Journal of Management Research and Analysis*, 2018, pp.2394-2770, Available: https://www.academia.edu/38493474/a_study_on_client_server_system_in_organizational_expectationS .

[7] B. Siregar, U. Andayani, N. Fatihah, L. Hakim, and F. Fahmi, “Tropical Timber Identification using Backpropagation Neural Network,” *Journal of Physics: Conference Series*, vol. 801, p. 012051, Jan. 2017, doi: https://doi.org/10.1088/1742-6596/801/1/012051

[8] P. Sahu, M. Singh, and D. Kulhare, “Implementation of Modified Hybrid Port Knocking (MHPK) with Strong Authentication,” *International Journal of Computer Applications*, vol. 64, no. 22, pp. 31–36, Feb. 2013, doi: https://doi.org/10.5120/10778-5478

[9] R. Muzawi, “Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk Optimalisasi Sistem Keamanan Jaringan Komputer,” *Sains dan Teknologi Informasi*, vol. 2, no. 1, pp. 52–58, 2016, doi: https://doi.org/10.33372/stn.v2i1.123.

[10] S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, and Z. Zhipeng, "Securing a Network: How Effective Using Firewalls and VPNs Are?" *Lecture Notes in Networks and Systems*, vol. 70, pp. 1050–1068, Feb. 2019, doi: https://doi.org/10.1007/978-3-030-12385-7_71.

[11] B. Pandey, G. A. Farulla, M. Indaco, L. Iovino, and P. Prinetto, “Design and Review of Water Management System Using Ethernet, Wi-Fi 802.11n, Modbus, and Other Communication Standards,” *Wireless Personal Communications*, vol. 106, no. 4, pp. 1677–1699, Feb. 2018, doi: https://doi.org/10.1007/s11277-018-5380-7.

[12] M. Shah, S. Ahmed, K. Saeed, M. Junaid, H. Khan, and Ata-ur-rehman, “Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning with Nmap Tool,” *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Jan. 2019, doi: https://doi.org/10.1109/icomet.2019.8673520.

[13] R. V. Deshmukh and K. K. Devadkar, “Understanding DDoS Attack & its Effect in Cloud Environment,” *Procedia Computer Science*, vol. 49, pp. 202–210, 2015, Doi: https://doi.org/10.1016/j.procs.2015.04.245.

[14] L. Bosnjak, J. Sres, and B. Brumen, “Brute-force and dictionary attack on hashed real-world passwords,” *2018 41st International Convention on*

Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2018, doi: https://doi.org/10.23919/mipro.2018.8400211.

[15] H. Hasdiana and H. Fahmi, "Aplikasi Pembelajaran Unified Modeling Language Berbasis Computer Assisted Instruction," *Query: Journal of Information Systems*, vol. 2, no. 2, Oct. 2018.

[16] H. E. Williams and D. Lane, "Web Database Applications with PHP and MySQL: Building Effective Database-Driven Web Sites", *O'Reilly Media, Inc*. 2004. Accessed: Jun. 21, 2024. [Online]. Available: https://books.google.iq/books?hl=ar&lr=&id=WuxxvP7RZasC&oi=fnd&pg=PR5&dq=Williams

[17] R. Wodyk and M. Skublewska-Paszkowska, "Performance comparison of relational databases SQL Server, MySQL, and PostgreSQL using a web application and the Laravel framework," *Journal of Computer Sciences Institute*, vol. 17, pp. 358–364, Dec. 2020, doi: https://doi.org/10.35784/jcsi.2279.

[18] M. Idhom, H. Wahanani, and A. Fauzi, "Network Security Applications Using the Port Knocking Method," *Journal of Physics: Conference Series*, vol. 1569, p. 022046, Jul. 2020, https://doi.org/10.1088/1742-6596/1569/2/022046..

[19] Z. Amir, S. Syaifuddin, and D. Risqiwati, "Implementasi Asymmetric Encryption Rsa Pada Port Knocking Ubuntu Server Menggunakan Knockd Dan Python," *Jurnal Repositor*, vol. 2, no. 6, p. 787, Apr. 2020, doi: https://doi.org/10.22219/repositor.v2i6.270.

[20] N. Iwan and D. Kurniawan, "Mikrotik Login Security with Port-Knocking and Brute Force Firewall at PT. Time Excelindo," *International Journal of Integrative Sciences*, vol. 2, no. 7, pp. 971–978, Jul. 2023, doi: https://doi.org/10.55927/ijis.v2i7.4782

[21] Y. Christian, "Analisis Sistem Pengamanan Akses Autentikasi Jaringan dengan Metode Port Knocking dan Action Tarpit pada Router Mikrotik," *Telcomatics*, vol. 4, no. 1, pp. 1–6, 2019, Accessed: Jan. 25, 2025. [Online]. Available: https://journal.uib.ac.id/index.php/telcomatics/article/view/586

[22] I. Marzuki, "Perancangan dan Implementasi Sistem Keamanan Jaringan Komputer Menggunakan Metode Port Knocking Pada Sistem Operasi Linux," *Jurnal Teknologi Informasi Indonesia (JTII)*, vol. 2, no. 2, pp. 18–24, Apr. 2019, doi: https://doi.org/10.30869/jtii.v2i2.312.

[23] T. Sanguankotchakorn and S. Kumar Arugonda, "Hybrid Controller for Securing SDN from Switched DDoS and ARP Poisoning Attacks," *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp.1–6, Sep 2019,doi: https://doi.org/10.23919/apnoms.2019.8893030.

[24] Y. Mulyanto, *et a*l., "Implementasi Port Knocking Untuk Keamanan Jaringan Smkn 1 Sumbawa Besar," *Jurnal Informatika Teknologi dan Sains*, vol. 3, no. 2, pp. 326–335, May 2021, doi: https://doi.org/10.51401/jinteks.v3i2.1016.

[25] P. Mehran, E. A. Reza, and B. Laleh, "SPKT: Secure Port Knock-Tunneling, an enhanced port security authentication mechanism," *IEEE Xplore*, Mar. 01, 2012. doi: https://doi.org/10.1109/ISCI.2012.6222683.

[26] M. Nur et al., "The Effectiveness of the Port Knocking Method in Computer Security," *International Journal of Integrative Sciences*, vol. 2, no. 6, pp. 861–868, Jun. 2023, doi: https://doi.org/10.55927/ijis.v2i6.4526.

[27] J. M. Parenreng, F. Rizal, and M. S. Wahyuni, "Simulation and Analysis of Network Security using Port Knocking and Intrusion Prevention System on Linux Server," *Internet of Things and Artificial Intelligence Journal*, vol. 4, no. 2, pp. 226–243, May 2024, doi: https://doi.org/10.31763/iota.v4i2.726.