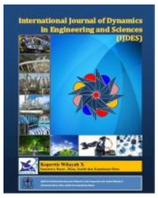
INTERNATIONAL JOURNAL of DYNAMICS in ENGINEERING and SCIENCES (IJDES)
LLDIKTI WILAYAH X
Vol. 8, No. 2, Oct 2023
e-ISSN: 2502-0692



IMPLEMENTATION INTRUSION DETECTION PREVENTION SYSTEM AS A SECURITY SYSTEM USING SNORT AND IPTABLES BASED ON LINUX

Ruri Hartika Zain [1,*], Yelmi Rahmawati[2]
[1,2,3] Universitas Putra Indonesia "YPTK" Padang

*Correspondence should be addressed to rurihartikazain@upiyptk.ac.id

Article Information

Submitted :
02 Agust 2023

Accepted :
20 Sept 2023

Published :
01 Oct 2023

Abstract

The development of computer networks continues, in terms of scalability, number of nodes, and technology. Computers connected to the network have the potential to experience disturbances or attacks. Therefore network security is very important in a computer network system to avoid attacks/disturbances and protect computer networks. Intrusion Detection System (IDS) with Snort implemented in the operating system linux can perform DoS attack monitoring (Denial of Service) and Port Scanning. Snort mode IDS will give alert regularly real-time according to rules Snort which is set in local.rules. IPTables as tools IPS will stop the attack/interference with rules IPTables applied. In this study, system testing was carried out Snort IDS, IPTables and service quality testing server. The results of the Snort IDS test can provide an alert that there is disturbances/attack real-time. IPS test results can overcome incoming attack/disturbances by blocking the intruder's IP address. Testing the quality of server service after implementing IDPS, the index value obtained was 3.75. Previously, server service quality had an index value of 2. This means that IDPS is able to overcome attacks/disturbances that enter the network.

Keywords: Network Security, IDS, Snort, Linux, DoS, Port Scanning, IPS, IPTables, Server Service Quality.

1. INTRODUCTION

The development of computer networks continues, in terms of scalability, number of nodes, and technology.

Computer network is an interconnection of two or more computers with wired or wireless transmission media. The term client-server is commonly used in computer networks. The client is the party requesting/receiving the service, while the server is the party providing/sending the service[1]. Devices with wireless transmission allow information to be sent between hosts without wires using electromagnetic waves[2].

The prosecutor's office which is the object of the author's research is a government agency that has a computer network infrastructure and server. In this research object, there is a problem, namely the absence of computer network monitoring. When a disturbance/attack enters the server at the prosecutor's office, the administrator does not know what type of disturbance/attack has entered the server. Attacks/disturbances that enter without knowledge and are not immediately handled to stop the incoming disturbance/attack can cause damage.

Common attacks or disturbances on computers connected to the network are DoS attacks and port scans. DoS attacks originate from a single device, whereas DDoS attacks are more than one device. DoS and DDoS are both traffic flooding attacks that use large data packets that can overwhelm and block access to servers. Commonly used attacks are UDP Flooding, SYN Flooding, and Ping of Death[3]. Port scan attacks are carried out by scanning the target network port, analyzing the target network port and then looking for gaps in the target port that are open[4]. Therefore network security is very important in a computer network system to avoid attacks and protect computer networks from external and internal network threats.

From these problems, an alternative solution is to implement network security using the IDPS method using Snort and IPTables. Snort IDS can detect attacks and IPS IPTables can perform filtering actions by inputting the attacker's IP address[5].

2. RESEARCH METHODS

The research method used is implementing the IDPS Intrusion Detection Prevention System using Snort and IPTables.

Intrusion Detection System (IDS) is a software or hardware application capable of detecting suspicious activity in a system or network. If suspicious activity is detected in network traffic, an Intrusion Detection System (IDS) will alert the system or administrator[6]. Meanwhile, the main function of an Intrusion Prevention System (IPS) is to stop an attack in progress[7].

Snort is a Linux system installation package tool that can detect intruders, analyze packets in real-time, and save log files to a database. Snort is an example of an IDS in the NIDS category that detects intrusion in network systems. Snort can work as a packet-logger to log network traffic and provide alerts, and as a packet sniffer to read network traffic. Snort is used as a detection and prevention tool for indicating a data packet in network traffic as threats. Snort also has rules like a firewall as a threat detector on the network. The implementation of the Snort application uses a rule set that allows Linux systems to detect and provide warnings against attack patterns from attackers[8].

IPTables is a tool that functions as a filter or data traffic regulator in the Linux operating system. IPTables has three types of rules in the filter table, namely firewall chains. There are three chains, namely INPUT, OUTPUT, and FORWARD. IPTables has three tables, namely, NAT, MANGLE, and FILTER. Filters function as data packet filters, such as DROP, LOG, ACCEPT or REJECT. NAT functions as a substitute for the origin or destination address of the data packet. Mangle functions to refine data packets such as TTL, TOS, and MARK. RAW is used to configure exceptions from connection tracking with NOTRACK[7].

Linux is an Open Source operating system based on GNU/Linux with various variants such as Slackware, Linux Mint, Debian, Open Suse, Archlinux, Redhat, and other Open Source software. Many variants of GNU/Linux only provide certain applications which may be of little use to the user. This resulted in many users remastering to meet their needs[9]. Linux installation is done in the virtualbox application to minimize the risk of failure. VirtualBox itself is a program for computer virtualization on desktop computers, servers, and laptops. Can virtualize 32-bit and 64-bit operating systems on computers with Intel and AMD processors in both software and hardware. Virtualbox is a free and open source virtualization software that provides the convenience and ability to create virtual machines natively[10].

3. RESULTS AND DISCUSSION

Implementation is done by installing on the IDPS server using Linux Mint, installing Snort software, Snort configuration and Snort rules. To overcome attacks/disturbances, configure IPTables rules, which are IPS tools. After the system has been successfully installed as a whole, then testing is carried out, tests are performed to prove that the implemented system can work properly. The test that the author will try to do to test the security system is to attack SYN flood and scan port.

A. IDS Testing With SYN Flood Attack

At this stage, the author tries to attack SYN Flood through the Backtrack 5 terminal. SYN Flood is one of the Denial of Service (DoS) and Distributed Denial 0f Service (DDoS) attacks where this attack aims to consume resources from the server so that the server cannot serve network traffic that is really legitimate.

The following is a look at an attack or disturbances performed using *Backtrack 5*:

Figure 1 *SYN Flood* attack from *Backtrack* 5 (1)

Figure 2 SYN Flood attack from Backtrack 5 (2)

In figures 1 and 2 the Syn Flood attack is carried out with the command *hping3 -i u1 -S -p 80 192.168.1.2,* which is launched simultaneously through two operating systems running on a virtual machine. Figure 3 is a Snort IDS alert set in local.rules.

Figure 3 SYN Flood *Attack/* Disturbances *Alerts*

From the image above it can be seen that the implemented Snort IDS can run well. Snort IDS installed on the server can detect interference that enters the server that has been implemented Snort IDS. From the capture results show information that IP 192.168.1.6 and IP 192.168.1.7 which are IP intruder perform a Syn flood against IP 192.168.1.2 which is the IDPS server IP with a warning *"Warning! SYN Flooding!"*. Complete with information on time, date of incident and classification of attack/disturbance.

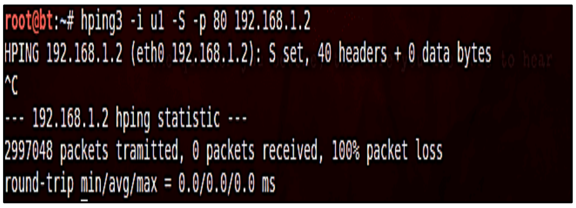


Figure 4 *SYN Flood* Attack from *Backtrack 5* (1) Blocked

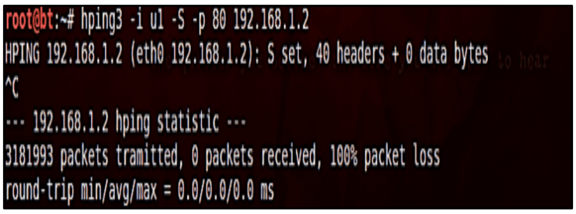


Figure 5 *SYN Flood* attack from *Backtrack 5* (2) Blocked

The results shown in figures 4 and 5 state that the two *intruders* cannot perform a *SYN Flood attack* to IP address 192.168.1.2 which is the *server IP, which means that the IPTables rules* that have been applied have successfully stopped the attack.

B. IDS and IPTables Testing With Ping of Death Attacks

The next test performs an attack from Windows 10 to the server with the method of requesting a reply from the server repeatedly intending to keep the server machine busy responding to requests from intruders. This attack attempt was carried out via the Windows 10 command prompt.

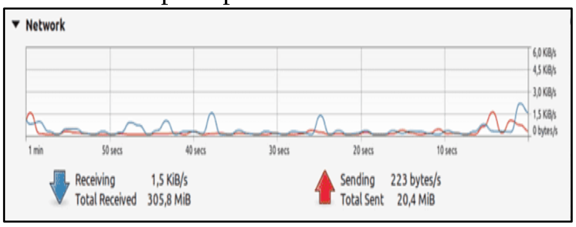


Figure 6 Display *of server resources* before an outage

Before the Ping of Death attack, the network graph display displayed by the IDPS server resource still looked normal, there was no significant increase in the network graph.

The following shows a picture of the Ping of Death attack from the Windows 10 command prompt:

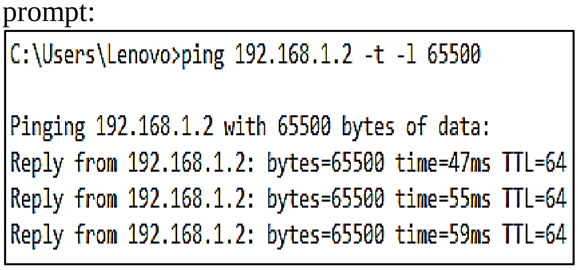


Figure 7 Display Ping of Death Attack

The icmp attack is performed with a load count of 65500. The intruder continuously sends this load count to the server's IP address.

```
07/12-23:59:39.476947  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.5 -> 192.168.1.2
07/12-23:59:39.478320  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.5
07/12-23:59:40.504933  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.5 -> 192.168.1.2
07/12-23:59:40.506327  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.5
07/12-23:59:41.514565  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.5 -> 192.168.1.2
07/12-23:59:41.515684  [**] [1:10000004:4] Warning! Ping Of Death! [**] [Classification:
Potentially Bad Traffic] [Priority: 2] {ICMP} 192.168.1.2 -> 192.168.1.5
```

Figure 8 Display of Ping of Death Attack Alert

The capture shows information that IP 192.168.1.5 which is the IP of the intruder performs Ping of Death against IP 192.168.1.2 which is the IP of the IDPS server with the warning "Warning! Ping of Death!". Complete with time and date information.

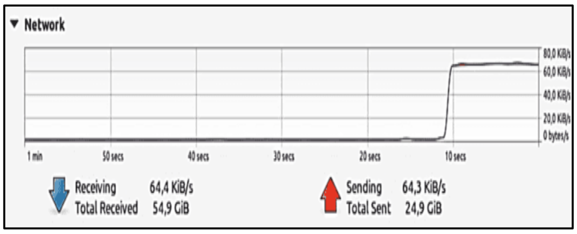


Figure 9 Display *of Resource Server* After Disruption

In figure 9 there is a significant increase in the *network* graph, packets received and sent increase to 64.4 KiB and 64.3 KiB per second from the same IP address.

```
C:\Users\Lenovo>ping 192.168.1.2 -t -l 65500

Pinging 192.168.1.2 with 65500 bytes of data:
Reply from 192.168.1.2: bytes=65500 time=44ms TTL=64
Reply from 192.168.1.2: bytes=65500 time=59ms TTL=64
Reply from 192.168.1.2: bytes=65500 time=43ms TTL=64
Request timed out.
Request timed out.
Request timed out.
```

Figure 10 Display of attacks after IPTables rules are applied

The result shown in figure 10 is the result of a Ping of Death attack after applying the IPS rules of Iptables. The results obtained state that the IP block using the IPTables rule was successful. It can be seen in the picture that is Ping of Death to IP address 192.168.1.2 which is the IP address of the IDPS server that is running suddenly experiences a Request time out (RTO) on the intruder machine which means that the intruder machine cannot Ping of Death to the destination IP address.

C. IDS and *IPTables* Testing With Nmap Port Scan Attack

The next test is to carry out a UDP port scan attack/disturbance and a TCP port scan using the Zenmap available on Backtrack, to find which ports are on the IDPS server.

a) TCP Port Scan

The following image shows the TCP port scan attack through Zenmap Backrtack:

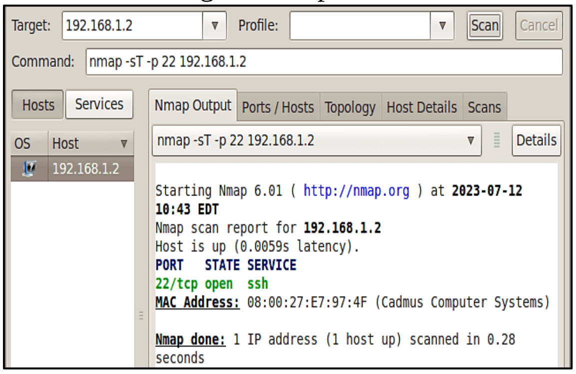


Figure 11 Display of *Nmap* tests with TCP protocol

Can be seen in Figure 11, the author tries to do a TCP port scan using Zenmap to IP address 192.168.1.2. The results of the image state that there is a port with the TCP protocol that is open on the IDPS server machine, namely port 22.

```
root@tokkie:~/snort_src/snort-2.9.20# snort -A console -q -i enp0s3 -c /etc/snort/snort.conf
07/13-00:04:21.803328  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
07/13-00:04:21.808153  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
07/13-00:04:21.808153  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
```

Figure 12 Display *Nmap Attack Alerts* with TCP Protocol

Figure 12, showing an alert view of Snort IDS. It can be seen that the intrusion originated from IP 192.168.1.12 with TCP protocol to IP address 192.168.1.2 which is the IDPS server machine. IDPS with the warning *"Warning! NMAP TCP Scan!"*. In the Snort IDS capture results above, there is also the time, date of the incident and classification of the attack.

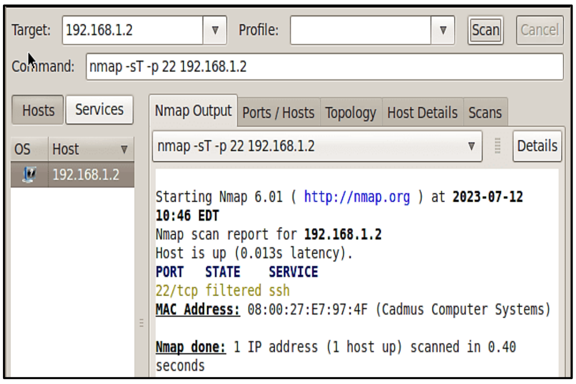


Figure 13 Testing *Blocked TCP Protocol* Nmap

Figure 13 states that the intruder cannot perform a port scan to IP address 192.168.1.2, it can be seen from the table that port 22 with the TCP protocol has been filtered.

b) UDP Port Scan

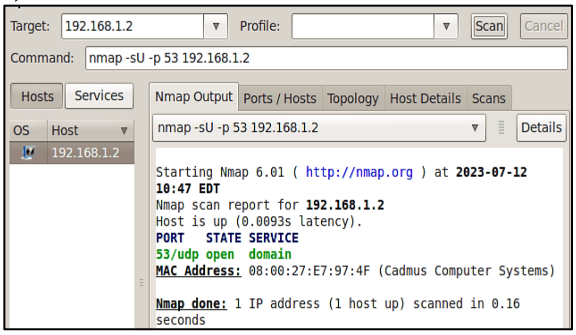


Figure 14 Display of *Nmap* tests with UDP protocol

Can be seen in Figure 14, the author tries to do a UDP port scan which also uses Zenmap to IP address 192.168.1.2. The results of the image state that there is a port with the UDP protocol that is open on the IDPS server machine, namely port 53.

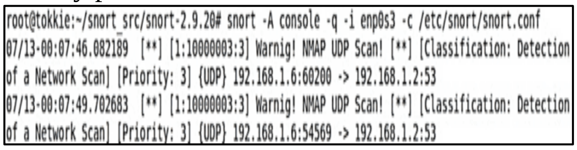


Figure 15 Display *Nmap Attack Alerts* with UDP Protocol

In figure 15, the *alert display from* Snort *IDS there is interference coming from the* same IP address, namely 192.168.1.13 with UDP protocol to IP address 192.168.1.2 with the warning "Warning! *NMAP UDP Scan!*". The *Snort* IDS capture results above show the time and date of the incident.

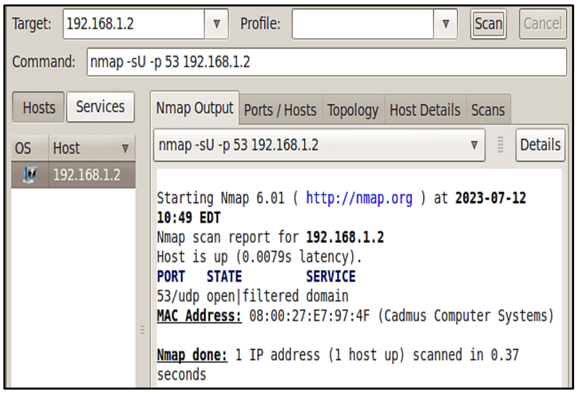


Figure 16 *Nmap* Testing of Blocked UDP Protocols

Figure 16 explains that the intruder cannot perform a port scan to IP address 192.168.1.2, it can be seen from the table that port 53 with the UDP protocol has been filtered.

D. Server *Service Quality Testing*

Measuring the quality of server service in this study the author used a QoS table. QoS (Quality of Service) is a measurement method used to determine the capabilities of a network. This QoS certainly already has a standardized assessment of TIPHON. Tests were conducted using iperf3, ping and wireshark. Iperf3 is used to test the upload and download speed of a server. Ping is done to see how many packets are lost. Then Wireshark is used to determine throughput, delay and jitter.

a) Testing Before a Syn Flood Attack

Upload	*Download*
94,0 Mbit	95,2 Mbit

Tabel 1 Download upload *speed* before an attack

Upload downloads tested with iperf3 client to the server, the value obtained is 94.0 Mbit for upload and 95.2 Mbit for download.

Parameter QoS	Value Average	Index	Category
Throughput	83 Mbit	4	Excellent
Delay	0,50ms	4	Excellent
Jitter	0,03ms	4	Excellent
Packet Lost	0%	4	Excellent
Average Index		**4**	**Excellent**

Tabel 2 QoS parameters before an attack

The results obtained in table 2 are the results of the *iperf3 test* and the *ping test* from the *client*, each tested for ten seconds. The results obtained before the entry of disruptions, the quality of *server* service is in the very good category. *Uploads* and *downloads* owned are of great value. Because if *Upload* and *download* the greater the value, the better.

b) Testing when there is a Syn Flood Attack

Upload	*Download*
210 Kbit	1,46 Mbit

Tabel 3 Download Upload *Speed* When There Is an Attack

Parameter QoS	Value Average	Index	Category
Throughput	10 Kbit	1	Bad
Delay	236,56ms	3	Good
Jitter	6,88ms	3	Good
Packet Lost	70%	1	Bad
Average Index		**2**	**Medium**

Tabel 4 QoS parameters during an attack

Tests conducted at the time of an attack, the *server* is in the medium category. Because the average index obtained at that time was 2, it means that the quality of *server services* decreased by 50%. The value of *uploads* and *downloads* also decreased dramatically.

c) Testing When the Syn Flood Attack Is Resolved

Upload	***Download***
86,0 Mbit	91,4 Mbit

Tabel 5 Download Upload Speed When the Attack Is Resolved

Parameter QoS	Value Average	Index	Category
Throughput	68 Mbit	4	Excellent
Delay	0,71ms	4	Excellent
Jitter	1,13ms	3	Good
Packet Lost	0%	4	Excellent
Average Index		**3,75**	**Good**

Tabel 6 QoS parameters when the attack is resolved

Tests conducted when the attack has been successfully resolved with *IPTables rules*, the server improved again, although not 100%, but the quality of *server* service is in the good category.

4. CONCLUSION

The application of IDS Snort on the server can effectively work as an open source-based computer network security in detecting an attack or interference on the IDPS server machine. IPTables can be a solution to overcome interference or attacks that enter the IDPS server. By applying this method in the tests that researchers do, it can restore the quality of server services in a network with an index of 3.75 from an index value of 4.

References

[1] K. Al Fikri and Djuniadi, "Keamanan Jaringan Menggunakan Switch Port Security," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 2, pp. 302–307, 2021, doi: https://doi.org/10.30743/infotekjar.v5i2.3501.

[2] R. W. Ismail and R. Pramudita, "Metode Penetration Testing pada Keamanan Jaringan Wireless Wardriving PT . Puma Makmur Aneka Engineering Bekasi," *J. Mhs. Bina Insa.*, vol. 5, no. 1, pp. 53–62, 2020, [Online]. Available: https://ejournal-binainsani.ac.id/index.php/JMBI/article/view/1373

[3] Wahyuni and P. Adytia, "Perbandingan Algoritma Machine Learning Dalam Mendeteksi Serangan DDOS," *Temat. J. Teknol. Inf. Komun.*, vol. 9, no. 2, pp. 161–166, 2022, doi: 10.38204/tematik.v9i2.1070.

[4] N. Nuryadi and E. C. Nainggolan, "Implementasi Intrusion Detection System Pada Local Area Network (Studi Kasus : Yayasan Pendidikan Tanah Tingal Tangerang)," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 19, no. 1, pp. 1–8, 2021, [Online]. Available: https://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/11098

[5] G. Tambunan and I. Mantra, "Implementasi Keamanan Ids / Ips Dengan Snort Dan IP Tables pada Server," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia, 28 Januari 2020 IMPLEMENTASI*, pp. 10–16, 2020, [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/352

[6] D. Kusuma, U. Darussalam, and D. Hidayatullah, "Implementasi Monitoring Jaringan Melalui Aplikasi Sosial Media Telegram Dengan Snort," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 5, no. 1, pp. 6–9, 2020, doi: 10.37438/jimp.v5i1.242.

[7] H. Alamsyah, Riska, and A. Al Akbar, "Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 5, no. 1, p. 17, 2020, doi: 10.31328/jointecs.v5i1.1240.

[8] S. Khadafi, Y. D. Pratiwi, and E. Alfianto, "Keamanan Ftp Server Berbasiskan Ids Dan Ips Menggunakan Sistem Operasi Linux Ubuntu," *Netw. Eng. Res. Oper.*, vol. 6, no. 1, p. 11, 2021, doi: 10.21107/nero.v6i1.190.

[9] F. F. Phasa, J. D. Irawan, and S. A. Wibowo, "Sistem Autentifikasi Hostpot Menggunakan Ldap Server," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 4, no. 2, pp. 120–127, 2020, doi: 10.36040/jati.v4i2.2703.

[10] A. Z. Mardiansyah, Y. M. Abdussyakur, and A. H. Jatmika, "OPTIMASI PORT KNOCKING DAN HONEYPOT MENGGUNAKAN IPTABLES SEBAGAI KEAMANAN JARINGAN PADA SERVER," vol. 3, no. 2, pp. 189–199, 2021, doi: https://doi.org/10.29303/jtika.v3i2.144.