Research Article

Network Security System Implementation Using Intrusion Prevention System and Honeypot Technology at the Regional Revenue Office (Bapenda) of Padang City

[1,*]Rizky Rahmansyah, [2]Irzon meiditra

[1] PUI Educational and Technology, Universitas Prima Indonesia, Medan City, North Sumatra 20118, Indonesia
[2] Institut Teknologi Rokan Hilir, Rokan Hilir Regency, Riau 28953, Indonesia

* Corresponding Author: rizkyrahmansyah@unprimdn.ac.id

Abstract: The Regional Revenue Agency for the City of Padang is the executor of regional government administration in the area of regional revenue. Having many computers where each computer is connected to the internet and stores important data for the purposes of carrying out work activities, the data is often damaged, stolen, or attacked by irresponsible parties. A network security system is needed with the aim of preventing an intruder attack. The author implements attack diversion techniques and detects attacks on the Padang City Regional Revenue Agency by using the Intrusion Prevention System (IPS) and Honeypot. Based on the results of research that has been done, Intrusion Prevention System (IPS) and Honeypot can be implemented to prevent attacks that will enter the server. After testing the Intrusion Prevention System (IPS) attack and the Honeypot successfully detects and drops the address of the attacker, the Honeypot will divert the attacker to a mock server on the server, and the network security system that has been built can work properly. The results of this study prove that the Intrusion Prevention System (IPS) and Honeypot can be implemented and can successfully detect, prevent, and divert attackers.

Keywords: network security, intrusion prevention system, honeypot, data protection, controlling attackers

Citation: Rahmansyah, R., & Meiditra, I. (2025). Network security system implementation using intrusion prevention system and honeypot technology at the Regional Revenue Office (Bapenda) of Padang City. *Iota*, 5(3), 2–25. https://doi.org/10.31763/iota.v5i3.1013

Academic Editor: Adi, P.D.P
Received: June 15, 2025
Accepted: July 21, 2025
Published: August 23, 2025

1. Introduction

IPS (Intrusion Prevention System) Snort is a server security system that can prevent attacks by examining and recording all data packets and recognizing packets with sensors. When an attack has been identified, IPS Snort will deny access (block) and record (log) all identified data packets. However, by only using IPS Snort, which can only check and record incoming attacks, Allert is deemed insufficient to secure a server by collaborating with other server security systems that are felt to make server network security better.

Honeypot Artillery which functions when a Hacker tries to penetrate through an open port, it can be detected as if the Hacker can penetrate the system, then Honeypot Artillery will provide information about who the attacker is and how the attacker can enter the Snort IPS system to be recorded in the database which can be seen in the web interface, Allert recorded in experiments that have been carried out in the database as many as 9453 on the TCP protocol as much as 9%, UDP as much (Aminanto & Sulistyo, 2020).

This research designs a Snort-based Intrusion Prevention System (IPS) system for real-time traffic analysis and IPTables, a basic IPS system integrated with Honeypot as a fake system to find out the techniques used by intruders on the SDN network architecture. IPS will detect attacks based on the rules applied, and if there is an attack, IPS will provide an alert to the Controller. The attack then, the IPS will provide an alert to the Controller, which the Controller will check the database and divert the attack traffic to the Honeypot. The results show that the accuracy rate has a result of 99.87%, the average detection speed for Port Scanning, Ping of Death, ICMP Flood, and TCP SYN Flood attacks are 1.207s, 1.045s, 1.047s, and 1.101s, respectively. Then, on the measurement of quality of services (QoS), the experimental results show that after the attack is diverted, there is an increase in Throughput value and a decrease in Packet Loss value, which results in no traffic accumulation on certain hosts.(Barends et al., 2022).

Snort is an open source Intrusion Detection System (IDS) that is widely used by network administrators as a system for monitoring networks and detecting intrusion attacks on the network. Snort's function as an intrusion detection system can be developed into an Intrusion Prevention System (IPS) by enabling Snort in inline mode with Data Acquisition (DAQ). Data Acquisition (DAQ) is a module in which there is a packet capture scheme from an interface. Snort identifies the data packet as an intrusion because the data packet pattern is the same as the Snort rule pattern that defines it as an intrusion. The log of the intrusion detection is stored as an alert. In this final project, the author will configure an Intrusion Prevention System (IPS) by running Snort in inline mode using the AFPACKET DAQ on the Linux Ubuntu operating system. The reason for choosing Linux Ubuntu as the operating system is that Ubuntu is a Linux operating system that is easy to use and develop according to the wishes of its users.(Suhendi & Cahyo, 2021)

The development of computer security is the Intrusion Prevention System (IPS), which combines firewall engineering methods and Intrusion Detection System (IDS). The result of this research is a technology that can be used to prevent attacks that will enter the local network, check and record all data packets, and recognize sensor packets. When the attack has been identified, IPS will deny access (block) and record (log) all identified data packets. So IPS acts as a firewall that will allow and block, combined with IDS, which can detect packets in detail. The existence of a network security system, the PDTI Unira Malang server, is safer and can avoid intrusion.(Wahyudi & Utomo, 2021)

OSSEC works like a firewall that can allow or block. Meanwhile, this cowrie honeypot works like a real server to trap attackers as if they had successfully attacked. The system has been designed to be able to handle attacks such as Port Scanning, SSH brute force, Man in the Middle (MITM) attack, and Distributed Denial of Service (DDoS). The results of the comparison of attacks with this confusion matrix, OSSEC integrated with Honeypot Cowrie, show a great level of accuracy against DDoS attacks. Based on logs, detection accuracy can reach a percentage of 100%.(Susanti et al., 2022). The Regional Revenue Agency is an implementing element of local government administration in the field of regional revenue led by a Head of Agency who is under and responsible to the Mayor through the Regional Secretary.

Head of the Agency who is under the responsibility of the Mayor through the Regional Secretary. Has many computers where each computer is connected to the internet network, on each computer at the Regional Revenue Agency (Bapenda) of Padang City stores important data where the data is data for the purposes of carrying out work activities of the Regional Revenue Agency (Bapenda) of Padang City, the data is often damaged, lost, stolen or attacked by irresponsible parties, so a network security system is needed to protect the data stored on each computer, because considering this company has become one of the companies that can compete in its field.

2. Theory

2.1 *History of the Padang City Regional Revenue Agency (Bapenda)*

In the early days before the establishment of the Regional Revenue Agency (Bapenda) of Padang City, it was originally named the Regional Revenue Agency (Dipenda) of

Padang City. Padang City is incorporated in one OPD, namely, DPKA Padang City. However, along with the need to improve government and organizational performance, the Padang City DPKA was split into two OPDs, namely BPKAD Padang City, which was formed by Regional Regulation Number 6 of 2015, and the Padang City Regional Revenue Agency (Dipenda), which was formed by Regional Regulation Number 5 of 2015.

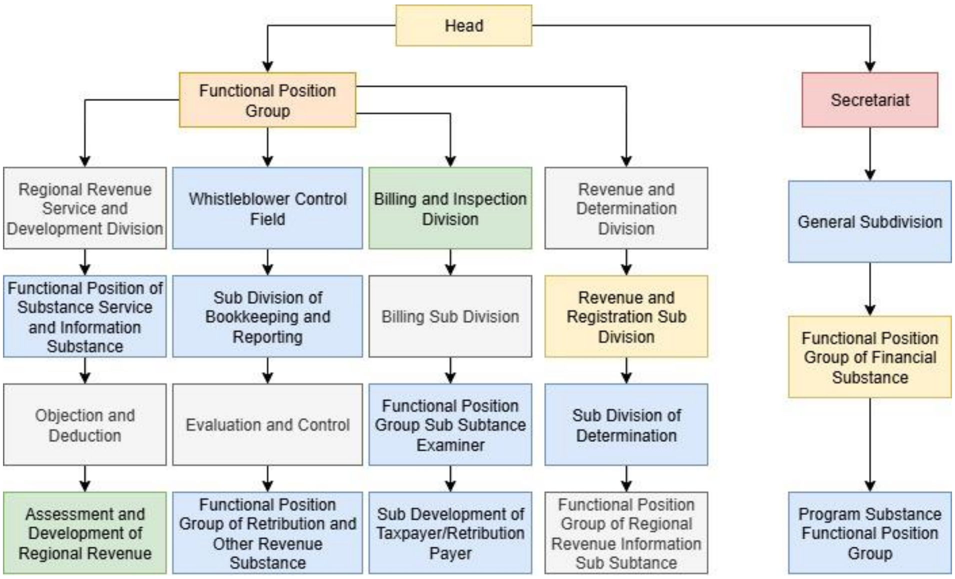

Head
Functional Position Group
Secretariat
Regional Revenue Service and Development Division
Whistleblower Control Field
Billing and Inspection Division
Revenue and Determination Division
General Subdivision
Functional Position of Substance Service and Information Substance
Sub Division of Bookkeeping and Reporting
Billing Sub Division
Revenue and Registration Sub Division
Functional Position Group of Financial Substance
Objection and Deduction
Evaluation and Control
Functional Position Group Sub Subtance Examiner
Sub Division of Determination
Assessment and Development of Regional Revenue
Functional Position Group of Retribution and Other Revenue Substance
Sub Development of Taxpayer/Retribution Payer
Functional Position Group of Regional Revenue Information Sub Subtance
Program Substance Functional Position Group

Figure 1. Organizational Structure of the Padang City Regional Revenue Agency

Then the Regional Revenue Agency of Padang City is an organizational restructuring carried out at the beginning of 2017 based on Regional Regulation No. 6 of 2016 concerning the Formation and Structure of Regional Apparatus of Padang City and Mayor Regulation No. 90 of 2016 concerning the position, organizational structure, duties, functions and work procedures of the Regional Revenue Agency, the Regional Revenue Agency (Dipenda) of Padang City changed its name to the Regional Revenue Agency (Bapenda) of Padang City. The Regional Revenue Agency is an implementing element of local government administration in the field of regional revenue led by a Head of the Agency who is under and responsible to the Mayor through the Regional Secretary.

2.2 *Information System: Definition and Implementation*

The definition of an information system is a system that provides information in such a way that it is useful to the recipient. In more detail, an information system can be defined as a set of entities consisting of hardware, software, and brainware that work together to provide processed data so that it is beneficial to the recipient of the data. An information system is a system within an organization that meets the needs of daily transaction processing, supports operations, managerial, and strategic activities of an organization, and provides certain external parties with the necessary reports. So, information system design is the development of a new system from an existing old system, where problems that occur in the old system are expected to be resolved in the new system. (Bastian et al. 2017)

Information systems are systems that can be defined by collecting, processing, storing, analyzing, and disseminating information for specific purposes. Information systems are complementary networks of hardware and software used by organizations to collect, filter, process, create, and distribute data. Information systems are a combination of hardware, software, and telecommunications networks that are built and used for information distribution and organizational arrangements, both commercial and non-commercial. An information system contains interrelated components that will control, analyze, visualize, and convey information to a centralized system in an organization. The components in this information system will determine the achievements and formulas for processing data into credible information. (Mufida et al. 2019)

Information Systems are an innovation that has previously occurred in human life and its evolution. Information Systems can also be interpreted as a branch of education that, in everyday life, always prioritizes the application of technology. The application of technology is taught starting from the basic level as a non-academic ability. Indirectly, humans must always follow the flow of globalization so that they are not left behind. Changes always occur in the social, economic, political, and even technological and cultural fields. Technology certainly does not only follow the flow of globalization, but is also proportional to the trends that occur, namely the trend of world currency exchange rates. If the exchange rate is high, the price of technology will also be high. (Nurlelah, Fuad 2022)

2.3 *Computer Network Security System*

A computer network security system is a system to prevent and identify unauthorized use of computer networks. The security system helps secure the network without hindering its use and raises anticipation when the network is successfully penetrated. (Riska et al. 2018) Computer network security is an issue that every computer user must pay attention to. It is necessary to clean up phishing sites, illegal links, spam, and so on on the computer. Never give opportunity to criminals because it is negligence that can have a serious impact on computer security. In addition, the development of computer network security technology should be continuously carried out as soon as possible, and reduce illegal elements technically.(Munawar et al. 2020)

Computer network security is very much needed during the rapid development of communication and network technology. This is to maintain the validity and integrity of data and ensure the availability of services for its users. The system must be protected from various attacks and intrusion attempts by third parties (attackers), and a system that is responsible for preventing and identifying anything unauthorized or suspicious in the network of a user's computer. The existence of a network security system means that all negative actions of intruders who want to access computers through a network system can be stopped or prevented. (Fachri and Harahap 2020)

2.4 *Computer Network Analysis*

Computer networks are telecommunications networks that allow computers to communicate with each other by exchanging data. Computer networks are built with a combination of hardware and software. When 2 or more computers communicate with each other or exchange data, there are actually parts of the computer network that are the parties that receive or request services, called clients, and those that provide or send them are called servers. This design is often called the Client-Server System. These interconnected computers must also have at least 1 network card each which is then connected via cable or wireless as a medium for data transmission and there is software for the network. as a data transmission medium and there is network operating system software that will form a simple computer network. If you want to create a wider range of computer networks, additional equipment is needed to support, such as Hubs, Switches, Routers, etc. (Astuti 2018).

A computer network is a structure consisting of computers, software and network devices that work together to achieve a predetermined goal. In order to achieve these goals, each part of the computer network receives and sends services. The party that uses the resources of the server is the client (client), and the one that provides various types of services is the server (server). This is called client-server and is commonly used in almost all computer network applications. (Tangkowit et al. 2021).

2.5 Honeypot

A honeypot is a way to trap or ward off unauthorized use attempts in an information system. Honeypot is a distraction for hackers, so that it looks as if they have succeeded in breaking into and retrieving data from a network, even though the data is actually not important, and the location has been isolated. (Laksana and Rosyid 2017) Honeypot is a technology to protect assets from misuse of information caused by cybercrime in recent years. Honeypots can also be used to improve corporate security detection. The Honeypot method is to lure attackers to interact and collect information that will be used for analysis. Based on the level of interaction

Honeypots can be divided into three types: low-interaction honeypot, medium-interaction honeypot, and high-interaction. In addition, Honeypot can detect and store attack information in the form of log data. Information stored in the data log includes attacker data in the form of IP addresses, ports attacked, services attacked, and attack times. Grouping attack data recorded in the form of log data is the right solution to determine the level of attack that can be detected.(Wibawa et al. 2020)

2.6 Low-Interaction Honeypot

Low Interaction Honeypot is a type of Honeypot that has characteristics that are easier and faster to implement. This is because this type of Honeypot only provides a clone of certain services. There is no real operating system that is used as a place of attack operations.(Fitriana and Khasanah 2018) Low Interaction Honeypot is a Honeypot that is designed to resemble the network infrastructure on the original server. Attackers are only able to check and connect to one or a few ports. A simple example of this type of Honeypot is the creation of a service that listens and records every connection that occurs on a port. Low Interaction Honeypot is a one-way connection because, from one side of the Honeypot, it only listens and records connections that occur without giving a reply to the connection. This will reduce the risk because there will be no system that will be taken over by the Low Interaction Honeypot Architecture. (Sari and Putra 2017)

2.7 High Interaction Honeypot

High Interaction Honeypot is a type of Honeypot that provides real systems and services like a real system; therefore, attackers can exercise full control over the Honeypot system. (Fitriana and Khasanah 2018) In a High Interaction Honeypot, there is an operating system where there is direct interaction with hackers, and there are no restrictions that limit this interaction. Eliminating these restrictions, the level of risk faced is higher because hackers can have root access. At the same time, the possibility of gathering information increases due to the high probability of attacks. The information can be in the form of attack patterns, programs used, motivations, and others. High Interaction Honeypot is recording and collecting more specific information than Low Interaction Honeypot, including the behavior of the attacker and the specific network protocol tracing of the attacker when an attack or intruder incident occurs. (Sari and Putra 2017)

2.8 Intrusion Prevention System

Intrusion Prevention System (IPS) is an approach that is often used by computer security systems. IPS combines firewall techniques and Intrusion Detection System (IDS) methods very well. This technology can be used to prevent attacks that will enter the local network by examining and logging all packets and recognizing packets with sensors. When the attack has been identified, the IPS will deny access (block) and log (log) all identified data packets. So, IPS acts like a firewall that will allow and block, combined with an IDS that can detect packets. Combined with an IDS that can detect packets in detail. IPS uses signatures to detect traffic activity on networks and terminals, where the detection of incoming and outgoing packets (inbound- outbound) can be prevented as early as possible before damaging or gaining access to the local network. (Arta et al. 2018)

An Intrusion Prevention System (IPS) is a type of network security method, either software or hardware, that can monitor unwanted activity or intrusion and can immediately react to prevent such activity. Intrusion Prevention System (IPS) is a development of the Intrusion Detection System (IDS). One of the methods or tools used as a security system on a server. IPS can provide security from an attack by utilizing the features of the IDS (Intrusion Detection System) and firewall as a feature to block access to network traffic. (Adesty et al. 2020)

2.9 Host-Based IPS (HIPS)

A Host-based Intrusion Prevention System is a prevention system consisting of many layers, using packet filtering, status inspection, and real-time prevention methods to keep the host in a state of proper performance efficiency. The mechanism works by preventing malicious code from entering the host and being executed without the need to check the threat signature. (Bossuyt et al. 2018) Host-based Intrusion Prevention System (HIPS) is the same as Host-Based Intrusion Detection System (HIDS). The HIPS agent program is installed directly on the protected system to monitor its internal system activities. HIPS is

HIPS binds to the operating system kernel and operating system services so that HIPS can monitor and intercept suspected system calls in order to prevent intrusion into the host. HIPS can also monitor data flow and activity in certain applications. For example, HIPS to prevent intrusion on a web server. On the security side, HIPS solutions may be able to prevent threats to the host. But in terms of performance, it must be considered whether HIPS hurts host performance. Because installing and binding HIPS to the operating system results in greater use of host computer resources.(Yoga Widya Pradipta 2017)

2.10 Network-Based IPS (NIPS)

Network-Based IPS (NIPS) or In-line proactive protection is able to hold all network traffic and inspect suspicious behavior and code. Because it uses an in-line model, high performance is a crucial element of IPS devices to prevent bottlenecks in the network. Therefore, NIPS is usually designed using three components to accelerate bandwidth performance. A network-based Intrusion Prevention System (NIPS) does not perform monitoring specifically on a single host. Instead, it monitors and protects the network globally. NIPS combines IPS features with firewalls and is sometimes referred to as In-Line IDS or Gateway Intrusion Detection System (GIDS). (Bossuyt et al. 2018) A network-based Intrusion Prevention System (NIPS) does not monitor specifically on one host. But monitoring and protection II-16 in a global network. NIPS combines IPS features with a firewall and is sometimes referred to as an In-Line IDS or Gateway Intrusion Detection System (GIDS). Popular IPS work systems are signature-based detection, anomaly-based detection, and file monitoring on the host operating system.(Yoga Widya Pradipta 2017)

2.11 Computer Network Terminology

Computer network terminology is the basics of forming a computer network, or it can also be called how a computer network is built and then implemented. A computer network is a collection of computers, printers, and other network equipment connected in one unit. The information and data move through a cable or wirelessly, which allows computer network users to communicate with each other. In network design, there are 3 common things in building a network, namely LAN, MAN, and WAN. (Simanjuntak et al. 2018)

3. Method

3.1 Research Framework

The research framework made in this research methodology has the aim that the steps taken by the author in carrying out the design do not deviate from the subject matter, and are easy to solve problems and are easy to understand. The steps that will be made are arranged systematically and can be used as guidelines. The sequence of steps in the research can be seen in Figure 2.

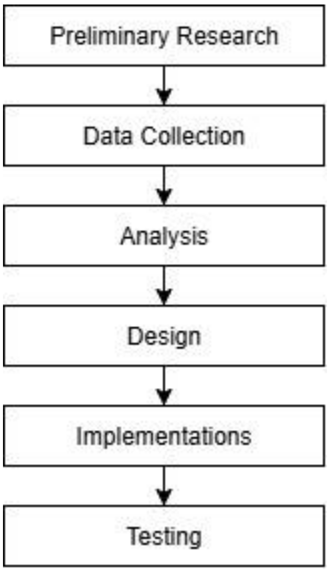


Figure 2. Research Framework

3.2 Research Step, Hardware, and Software

This research was conducted at the Regional Revenue Agency (Bapenda) of Padang City, which is located at Moh. Yamin Street No.70, Kp. Jao, Kec. Padang Barat, Padang City, West Sumatra. The research method used is qualitative, namely, developing network security at the Regional Revenue Agency (Bapenda) of Padang City. (Bapenda) Padang City. Collecting data and information for writing this research is done in several ways. Research conducted by going directly to the field research site to observe network services that are the object of research. This library research is carried out by reading, discussing, summarizing, and making conclusions from books, theories in libraries, and network security journals that have analysis on using ports and firewalls to obtain materials that can scientifically serve as a basis for preparing this research.

A research method carried out using personal computer (PC) tools. This research is carried out by designing programs or software that are in accordance with the topics and problems faced, and also in terms of preparing the report as a whole. The existence of data collection techniques, then the authors practice them and try to collect the necessary data as much as possible for the perfection of this final project. The specifications of the hardware used are Laptop Asus VivoBook X515MA, Intel Celeron N4020 Processor, 4 GB RAM, 256 GB SSD, and a Mouse. While the software used in this study includes Ubuntu 20.04 LTS Linux Operating System, Microsoft Office 2010, VMware Workstation 17 Player, Command Terminal, Snort, Honeypot, MobaXterm & PuTTY.

3.3 The Analysis Stage

The analysis stage is the stage of collecting information related to software development, including data, hardware, and system requirements. The analysis stage is carried out with the aim of finding the right solution to solve problems and avoid the emergence of new problems. The author analyzes the data that has been collected and analyzes the system that will be run as a solution to the problem formulation obtained. At this stage, data collection is carried out, which will later be processed, such as network architecture data, network topology systems, and other data that will complement the criteria for system development.

There are 2 methods used in this research, namely IPS and Honeypot. The IPS method is used to notify that there has been an attempted attack on the server, whose rules are made during the attack process. Assumes the attacker has successfully trapped in the Honeypot server and carried out several attacks on the server. All activities will be stored in the Honeypot log file. The script that is run on the Honeypot server retrieves some information and is recorded in a log. Based on the log, it is used as a reference to create several IPS rules, including the attacker's IP address, attacker port, port protocol service requested, and attacker URL request. This research also built and developed a program that analyzes and maps some content that contains illegal requests. Then it will create IPS rules automatically.

3.2 The Design Stage

This stage will build a server security system on the network; the author designs it to work as expected based on the analysis that has been obtained. The design is carried out using a network topology as a network security system design at the Regional Revenue Agency (Bapenda) of Padang City to explain the flow of analysis that will be made in conducting research. This stage is carried out by designing the network scheme that will be built in accordance with the topology that has been made. The server on the designed network is configured with IPS and Honeypot configuration on a network server using a terminal on Ubuntu Linux. Ubuntu Linux operating system. The design will be made in the form of a basic command display. In this implementation, researchers will discuss the network security system that will be used. This researcher will use IPS and Honeypot as network security tools for service servers.

3.3 Analysis and Design System

Analysis is a problem identification process consisting of some data that is worth information to build a system. Analysis is carried out to understand the problems of the system before configuring it. The analysis stage at Bapenda City of Padang on the existing network still uses the default from the router modem which allocates network sharing and in network security on the server at Bapenda City of Padang still uses a firewall contained in the system so that network security is still not optimal, and is vulnerable to attacks that try to enter through access that is not permitted by the attacker. So in this case, it is systematically organized and designed using software. Data analysis is the most important stage in the development of a system, which is the initial stage in the design and development of a security system. With data analysis, the needs and problems that exist will be identified so that improvements can be made to the system.

Implementing data in a security system requires data to be processed in order to implement the system. This research Data analysis consists of network architecture data, such as network topology and other data needed in the system and data analysis. The data obtained in this research comes from the object of research and some information obtained from direct interviews with sources. The data obtained in the form of general information about the object of research, ranging from company history, company structure, and information about the number of employees and staff in the company, as well as providing a clear picture of the architecture and network topology used by Bapenda Padang City. The network topology used at Bapenda City of Padang is shown in Figure 3.

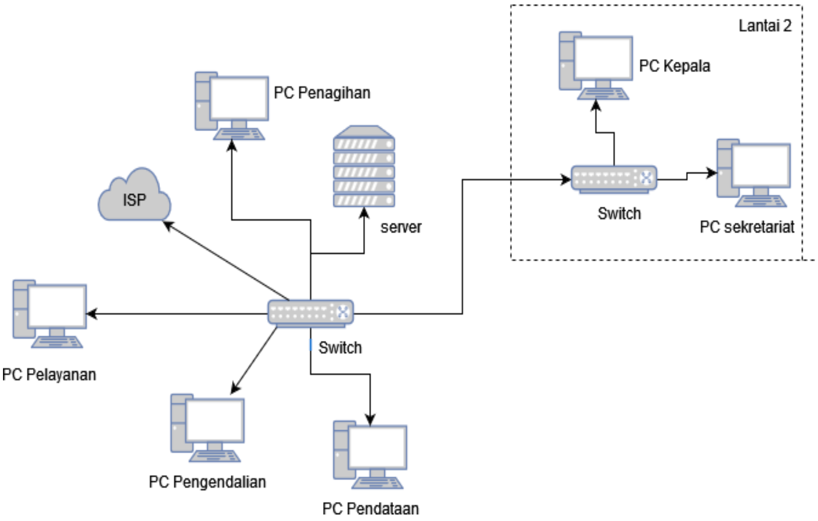
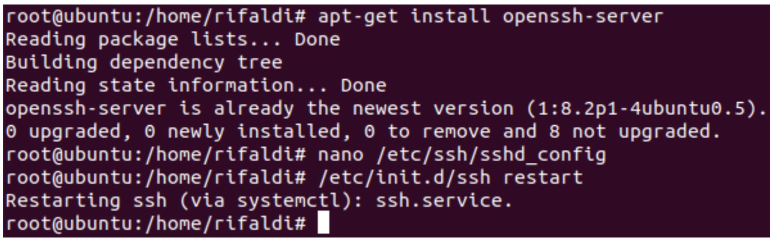


Figure 3. Network Topology of Bapenda City of Padang

4. Result and Discussion

4.1 SSH Server Configuration

Configure the ssh server which is the original server on the server network with port 22, install SSH on the Ubuntu operating system virtually on a vmware workstation just need to enter the apt-get install openssh- server command and therefore before installing cowrie and dependencies, change the default SSH port 22 to port 2222 in the sshd_config file so that attackers think that they are on the original SSH port and restart SSH to see if the newly configured port is being listened to. The installation of the SSH Server can be seen in Figure 4.

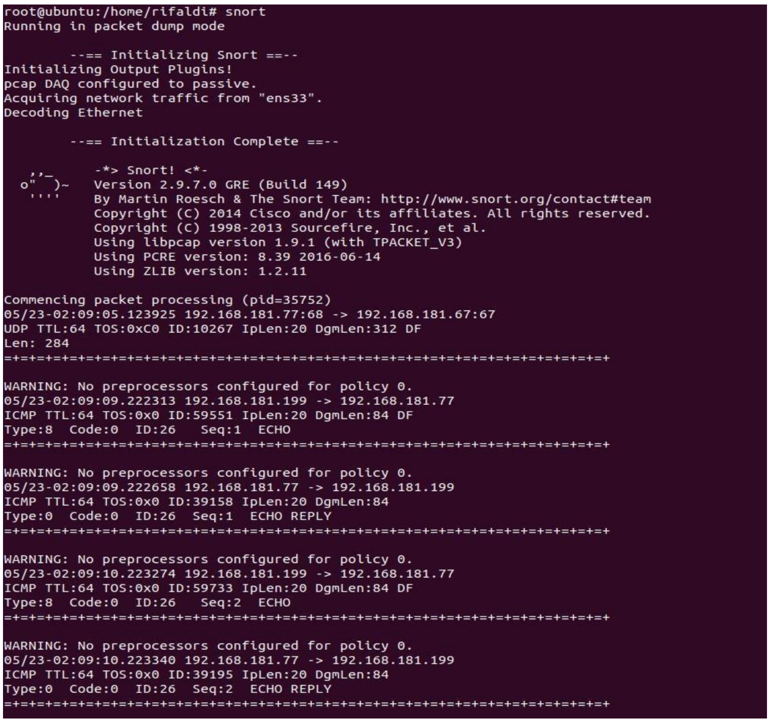


```
root@ubuntu:/home/rifaldi# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.5).
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
root@ubuntu:/home/rifaldi# nano /etc/ssh/sshd_config
root@ubuntu:/home/rifaldi# /etc/init.d/ssh restart
Restarting ssh (via systemctl): ssh.service.
root@ubuntu:/home/rifaldi# 
```

Figure 4. Installation of SSH Server

4.2 Snort Installation

Snort is an intrusion detection system that is useful for performing real-time packet logging and traffic analysis on IP networks. The author's Snort configuration uses the IPS program, and to install Snort, type the command sudo apt-get install snort. Snort is run, then the data packet will immediately appear and monitor the packet data sent or received through a particular network interface. If the network traffic is detected as a threat, then Snort will display an alert message as shown in Figure 5.



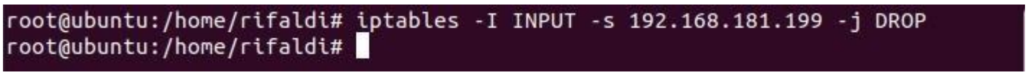
```
root@ubuntu:/home/rifaldi# snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=35752)
05/23-02:09:05.123925 192.168.181.77:68 -> 192.168.181.67:67
UDP TTL:64 TOS:0xC0 ID:10267 IpLen:20 DgmLen:312 DF
Len: 284
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/23-02:09:09.222313 192.168.181.199 -> 192.168.181.77
ICMP TTL:64 TOS:0x0 ID:59551 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:26   Seq:1  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/23-02:09:09.222658 192.168.181.77 -> 192.168.181.199
ICMP TTL:64 TOS:0x0 ID:39158 IpLen:20 DgmLen:84
Type:0  Code:0  ID:26  Seq:1  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/23-02:09:10.223274 192.168.181.199 -> 192.168.181.77
ICMP TTL:64 TOS:0x0 ID:59733 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:26   Seq:2  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/23-02:09:10.223340 192.168.181.77 -> 192.168.181.199
ICMP TTL:64 TOS:0x0 ID:39195 IpLen:20 DgmLen:84
Type:0  Code:0  ID:26  Seq:2  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 5. Snort Runs & Detects Intruders

Display Figure 5 shows that the Alert shows that someone from IP 192.168.181.199 successfully attacked the computer with IP Address 192.168.181.77. Furthermore, IPTables on a virtual server is a firewall that regulates all directions from the port that has been set. The IPTables system design is useful for preventing and collecting all attack packets on the network server so that the system can run according to a predetermined design. The IPTables installation process is carried out on the Linux Ubuntu 20.04 LTS Operating System. IPTables Installation. Before configuring IPTables, first install IPTables with the command :

```
# sudo apt-get install iptables-persistent
```

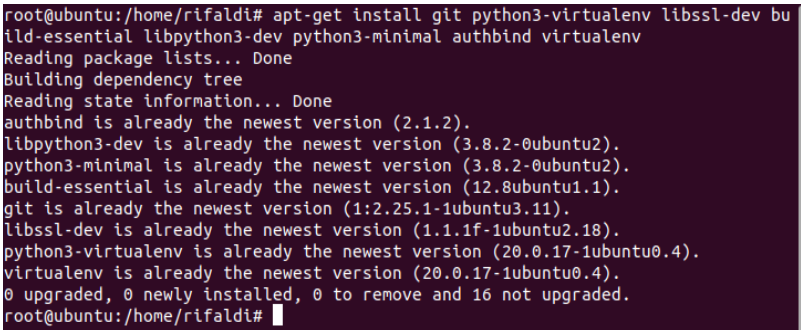
IPTables to prevent attacks on server ports. So to overcome attacks on networks that have been detected by IPS Snort, namely by blocking the attacker's IP Address, as shown in Figure 5.



```
root@ubuntu:/home/rifaldi# iptables -I INPUT -s 192.168.181.199 -j DROP
root@ubuntu:/home/rifaldi# █
```

Figure 5. Iptables Commands to Prevent Attacks.

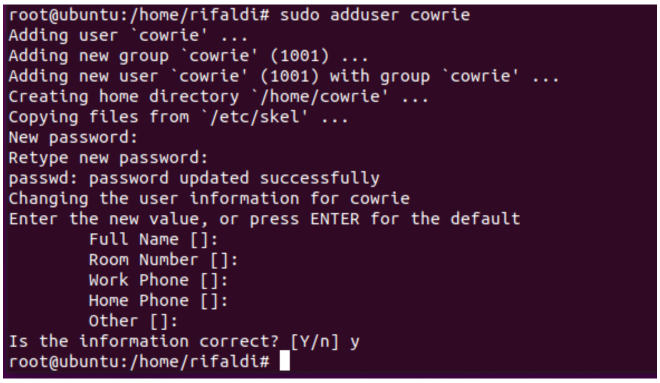
The Honeypot configuration phase installs system-wide support for the Python virtual environment and other dependencies. The actual Python package is installed later. So that a program can run properly and correctly.



```
root@ubuntu:/home/rifaldi# apt-get install git python3-virtualenv libssl-dev bu
ild-essential libpython3-dev python3-minimal authbind virtualenv
Reading package lists... Done
Building dependency tree
Reading state information... Done
authbind is already the newest version (2.1.2).
libpython3-dev is already the newest version (3.8.2-0ubuntu2).
python3-minimal is already the newest version (3.8.2-0ubuntu2).
build-essential is already the newest version (12.8ubuntu1.1).
git is already the newest version (1:2.25.1-1ubuntu3.11).
libssl-dev is already the newest version (1.1.1f-1ubuntu2.18).
python3-virtualenv is already the newest version (20.0.17-1ubuntu0.4).
virtualenv is already the newest version (20.0.17-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
root@ubuntu:/home/rifaldi# 
```

Figure 6. Install All Cowrie Dependencies

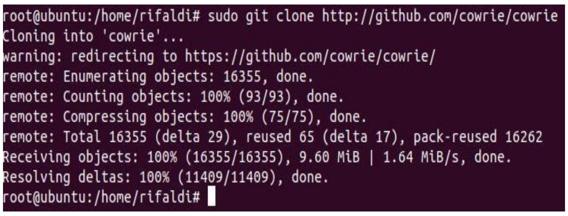
Creating a new user is, for obvious security reasons, not recommended as it does not allow Cowrie to run as root and creates a cowrie user with fewer privileges. Proses installasi All Cowrie Dependencies Meanwhile, the process of creating a new user is shown in Figure 7.



```
root@ubuntu:/home/rifaldi# sudo adduser cowrie
Adding user `cowrie' ...
Adding new group `cowrie' (1001) ...
Adding new user `cowrie' (1001) with group `cowrie' ...
Creating home directory `/home/cowrie' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for cowrie
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@ubuntu:/home/rifaldi# 
```

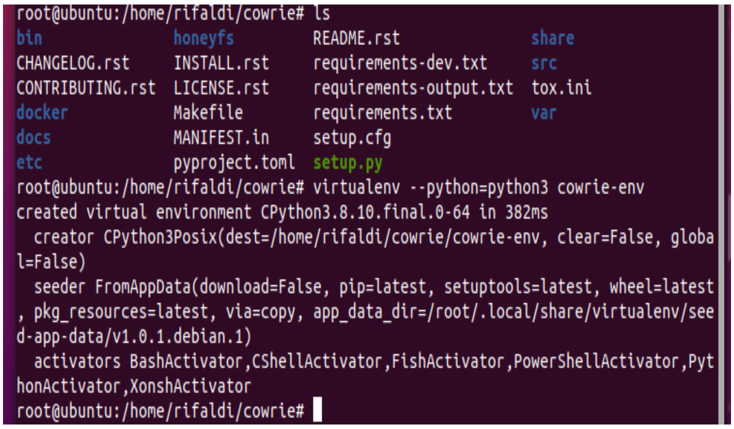
Figure 7. Create a New User

Clone the user cowrie that has been created using GitHub, as shown in Figure 8. The clone user cowrie stage is to install Cowrie from source code because it has to download Cowrie from the release page, extract it, and then run the installation command.



```
root@ubuntu:/home/rifaldi# sudo git clone http://github.com/cowrie/cowrie
Cloning into 'cowrie'...
warning: redirecting to https://github.com/cowrie/cowrie/
remote: Enumerating objects: 16355, done.
remote: Counting objects: 100% (93/93), done.
remote: Compressing objects: 100% (75/75), done.
remote: Total 16355 (delta 29), reused 65 (delta 17), pack-reused 16262
Receiving objects: 100% (16355/16355), 9.60 MiB | 1.64 MiB/s, done.
Resolving deltas: 100% (11409/11409), done.
root@ubuntu:/home/rifaldi# 
```

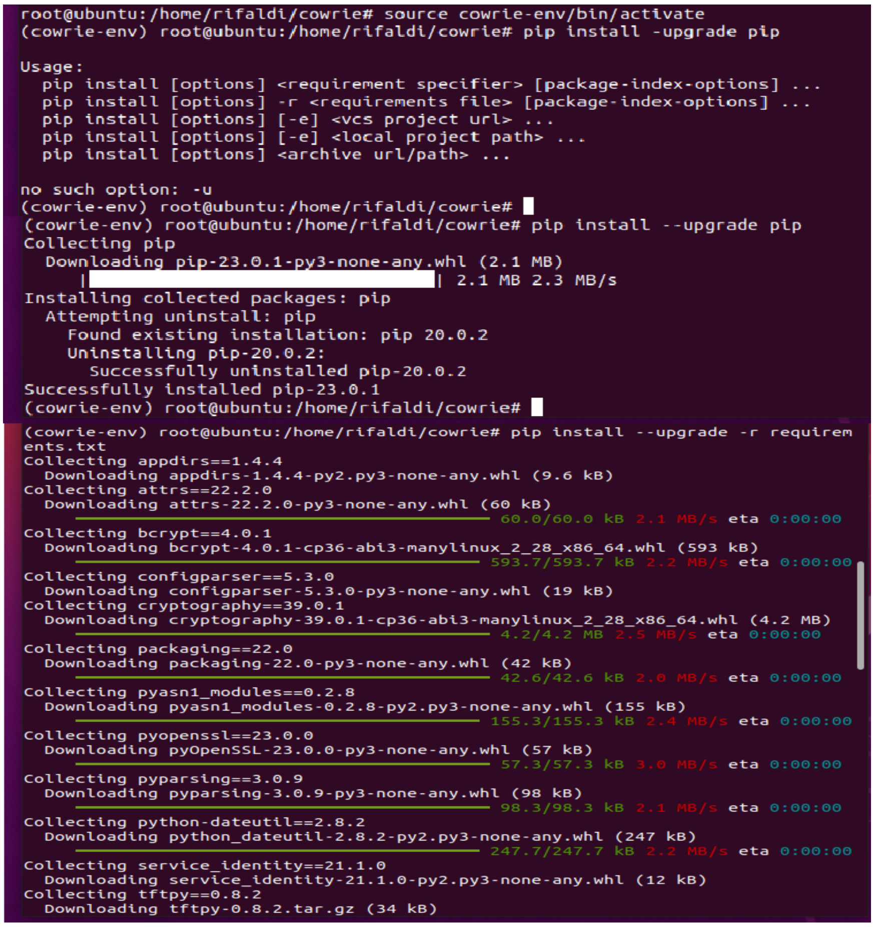
Figure 8. Cowrie User Clone Process Using GitHub



```
root@ubuntu:/home/rifaldi/cowrie# ls
bin              honeyfs        README.rst               share
CHANGELOG.rst    INSTALL.rst    requirements-dev.txt     src
CONTRIBUTING.rst LICENSE.rst    requirements-output.txt  tox.ini
docker           Makefile       requirements.txt         var
docs             MANIFEST.in    setup.cfg
etc              pyproject.toml setup.py
root@ubuntu:/home/rifaldi/cowrie# virtualenv --python=python3 cowrie-env
created virtual environment CPython3.8.10.final.0-64 in 382ms
  creator CPython3Posix(dest=/home/rifaldi/cowrie/cowrie-env, clear=False, globa
l=False)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest
, pkg_resources=latest, via=copy, app_data_dir=/root/.local/share/virtualenv/see
d-app-data/v1.0.1.debian.1)
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,Pyt
honActivator,XonshActivator
root@ubuntu:/home/rifaldi/cowrie# 
```

Figure 9. Virtualenv Settings on User Cowrie

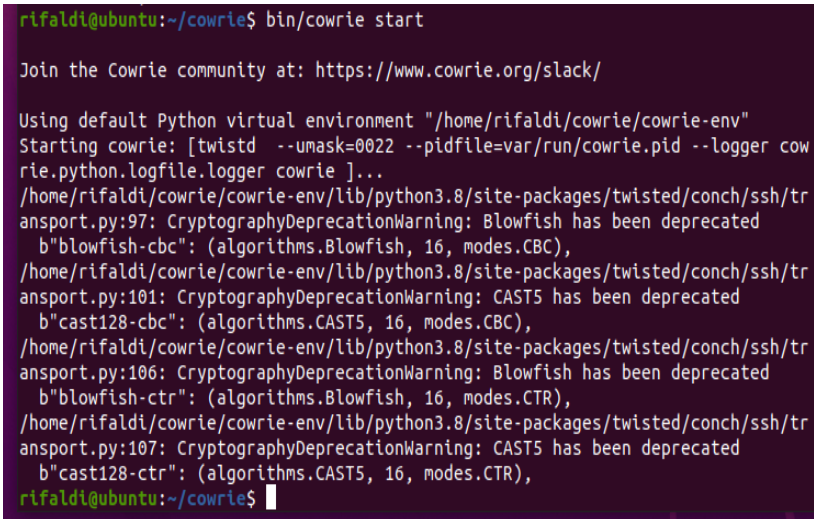
Enable virtual-env and install the package. Pip is used to install, upgrade, and remove Python packages. It is also used to manage the Python virtual environment. The requirements.txt file included with Cowrie is used as a reference for Python dependencies for pip to be installed. It will then upgrade pip from within the virtual environment and install all the requirements for Cowrie, as shown in Figures 9 and 10.



```
root@ubuntu:/home/rifaldi/cowrie# source cowrie-env/bin/activate
(cowrie-env) root@ubuntu:/home/rifaldi/cowrie# pip install -upgrade pip

Usage:
  pip install [options] <requirement specifier> [package-index-options] ...
  pip install [options] -r <requirements file> [package-index-options] ...
  pip install [options] [-e] <vcs project url> ...
  pip install [options] [-e] <local project path> ...
  pip install [options] <archive url/path> ...

no such option: -u
(cowrie-env) root@ubuntu:/home/rifaldi/cowrie# 
(cowrie-env) root@ubuntu:/home/rifaldi/cowrie# pip install --upgrade pip
Collecting pip
  Downloading pip-23.0.1-py3-none-any.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 2.3 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.0.2
    Uninstalling pip-20.0.2:
      Successfully uninstalled pip-20.0.2
Successfully installed pip-23.0.1
(cowrie-env) root@ubuntu:/home/rifaldi/cowrie# 
(cowrie-env) root@ubuntu:/home/rifaldi/cowrie# pip install --upgrade -r requirem
ents.txt
Collecting appdirs==1.4.4
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting attrs==22.2.0
  Downloading attrs-22.2.0-py3-none-any.whl (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.0/60.0 kB 2.1 MB/s eta 0:00:00
Collecting bcrypt==4.0.1
  Downloading bcrypt-4.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (593 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 593.7/593.7 kB 2.2 MB/s eta 0:00:00
Collecting configparser==5.3.0
  Downloading configparser-5.3.0-py3-none-any.whl (19 kB)
Collecting cryptography==39.0.1
  Downloading cryptography-39.0.1-cp36-abi3-manylinux_2_28_x86_64.whl (4.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.2/4.2 MB 2.5 MB/s eta 0:00:00
Collecting packaging==22.0
  Downloading packaging-22.0-py3-none-any.whl (42 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.6/42.6 kB 2.0 MB/s eta 0:00:00
Collecting pyasn1_modules==0.2.8
  Downloading pyasn1_modules-0.2.8-py2.py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.3/155.3 kB 2.4 MB/s eta 0:00:00
Collecting pyopenssl==23.0.0
  Downloading pyOpenSSL-23.0.0-py3-none-any.whl (57 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.3/57.3 kB 3.0 MB/s eta 0:00:00
Collecting pyparsing==3.0.9
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 2.1 MB/s eta 0:00:00
Collecting python-dateutil==2.8.2
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.7/247.7 kB 2.2 MB/s eta 0:00:00
Collecting service_identity==21.1.0
  Downloading service_identity-21.1.0-py2.py3-none-any.whl (12 kB)
Collecting tftpy==0.8.2
  Downloading tftpy-0.8.2.tar.gz (34 kB)
```

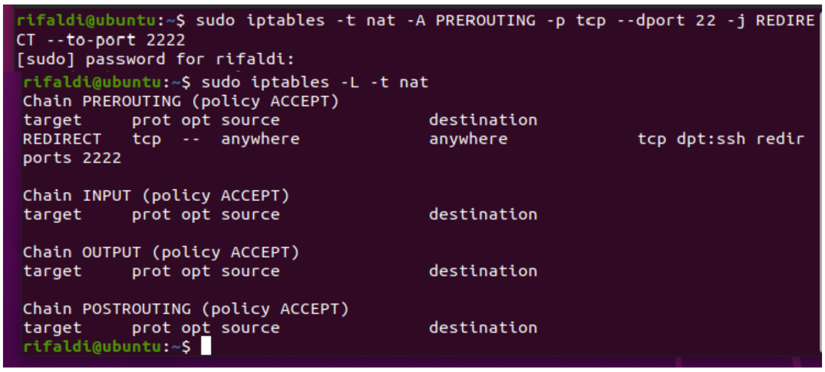
Figure 10. Enabling Virtual-Env and Installing Package on User Cowrie

Furthermore, start Cowrie with the cowrie command. Can add the cowrie/bin directory to the cowrie path if desired. Existing virtual environments are preserved if enabled; otherwise, Cowrie will try to load an environment called "cowrie-env". Port 22 needs to be redirected to port 2222 of the Cowrie honeypot SSH using Iptables in the Ubuntu operating system so that attackers who access the default port can be redirected to the Honeypot server. The port redirection command applies to the entire system and needs to be run as root, as shown in Figures 11 and 12.



```
rifaldi@ubuntu:~/cowrie$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using default Python virtual environment "/home/rifaldi/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cow
rie.python.logfile.logger cowrie ]...
/home/rifaldi/cowrie/cowrie-env/lib/python3.8/site-packages/twisted/conch/ssh/tr
ansport.py:97: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/rifaldi/cowrie/cowrie-env/lib/python3.8/site-packages/twisted/conch/ssh/tr
ansport.py:101: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/home/rifaldi/cowrie/cowrie-env/lib/python3.8/site-packages/twisted/conch/ssh/tr
ansport.py:106: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/home/rifaldi/cowrie/cowrie-env/lib/python3.8/site-packages/twisted/conch/ssh/tr
ansport.py:107: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
rifaldi@ubuntu:~/cowrie$ 
```

Figure 11. Enabling the Cowrie SSH Honeypot



```
rifaldi@ubuntu:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRE
CT --to-port 2222
[sudo] password for rifaldi:
rifaldi@ubuntu:~$ sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:ssh redir
ports 2222

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
rifaldi@ubuntu:~$ 
```

Figure 12. Redirect Port 22 Menuju Port 2222 SSH Honeypot Server- Bapenda

4.3 *System Test*

The testing stage of the Honeypot security system requires MobaXterm, which is an open-source tool that can be used to perform SSH, Telnet, and Rlogin network protocols. The following is an explanation of the use of supporting applications to test the results of the implementation of MobaXterm. MobaXterm is a tool that can be used for remote computing. MobaXterm has provided various remote tools that are important to use. For example, SSH, VNC, FTP, Telnet, and the like. It also provides Unix commands on the Windows Desktop. Of course, it's all in the form of one portable exe file. The MobaXterm Software Display is shown in Figure 13.

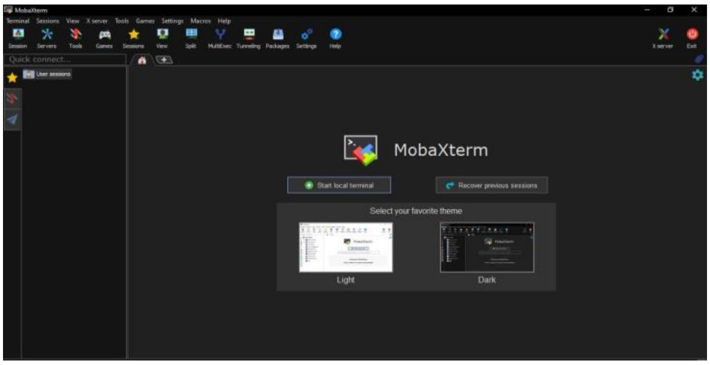
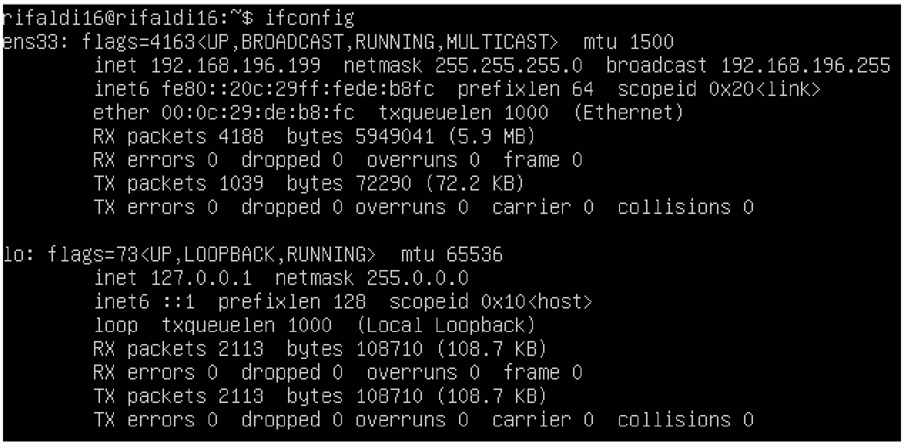

MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
User sessions
MobaXterm
Start local terminal
Recover previous sessions
Select your favorite theme
Light
Dark

Figure 13. MobaXterm Software Display

PuTTY is an open-source application that is often used to perform remote access, such as RLogin, SSH, and Telnet. Remote access is an application used to control a system from a distance or in a different place. Remote access is still connected to the internet. Server owners mostly use this application to access their servers. The remote location of the server makes PuTTY very useful, because there is no need to come directly to the server location to configure it. This stage tests the security system using the Intrusion Prevention System (IPS) and Honeypot. System testing aims to see whether the system designed is appropriate, or if there is damage or errors that occur when the implemented system is running. There are many denial of service attacks that are used to paralyze server networks. One type of attack that often occurs as an intrusion into the system can be a ping of death, which can make the network unstable, such as a network drop, freeze, crash, or lag (hang) on the server.

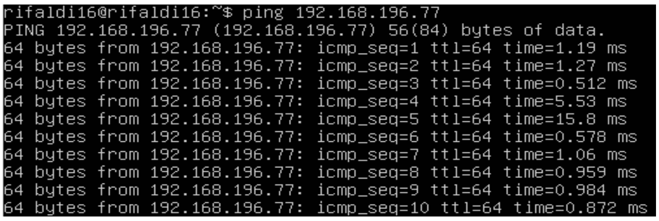
This test phase pings from client to server on the same network. Ping is a program used to check network activity based on technology (TCP / IP). Using this program, it can be tested whether a computer is connected to another computer. It works by sending a packet to the IP address to be tested for connection and waiting for a response from it. In this simulation, the attacker uses an Ubuntu operating system laptop with 4 GB RAM, connected to the same network as the server, and has an IP configuration of 192.168.196.199 as a client.



```
rifaldi16@rifaldi16:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.196.199  netmask 255.255.255.0  broadcast 192.168.196.255
        inet6 fe80::20c:29ff:fede:b8fc  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:de:b8:fc  txqueuelen 1000  (Ethernet)
        RX packets 4188  bytes 5949041 (5.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1039  bytes 72290 (72.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2113  bytes 108710 (108.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2113  bytes 108710 (108.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 14. IP Display of the Attacker

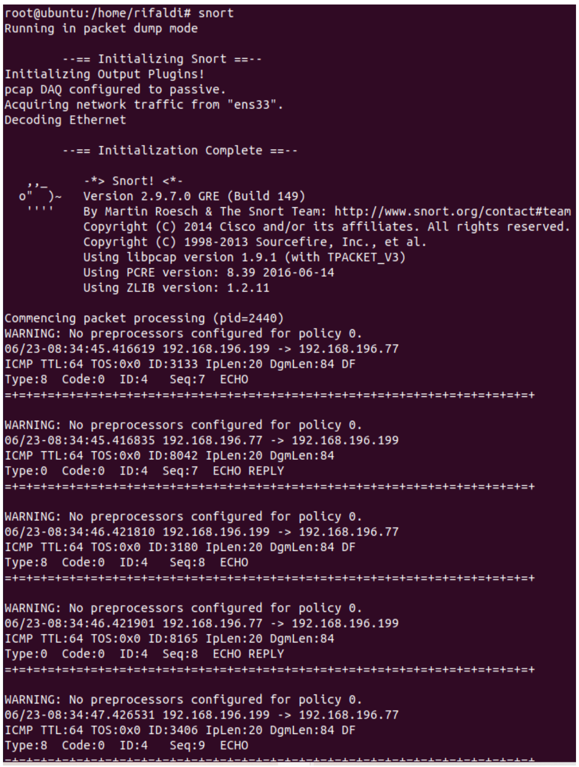
Ping works by sending a data packet called an Internet Control Message Protocol (ICMP) echo request. The way it works is that when pinging the target site (object) it will appear on the screen of the returned results in the form of IP number information from which ping gets an echo reply. The time it takes for the ping program to get the last reply is the laptop's time to live (TTL). This ping test was carried out using the Ubuntu Server operating system laptop with the ping 192.168.196.77 command in CMD from the client with IP address 192.168.196.199.



```
rifaldi16@rifaldi16:~$ ping 192.168.196.77
PING 192.168.196.77 (192.168.196.77) 56(84) bytes of data.
64 bytes from 192.168.196.77: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.196.77: icmp_seq=2 ttl=64 time=1.27 ms
64 bytes from 192.168.196.77: icmp_seq=3 ttl=64 time=0.512 ms
64 bytes from 192.168.196.77: icmp_seq=4 ttl=64 time=5.53 ms
64 bytes from 192.168.196.77: icmp_seq=5 ttl=64 time=15.8 ms
64 bytes from 192.168.196.77: icmp_seq=6 ttl=64 time=0.578 ms
64 bytes from 192.168.196.77: icmp_seq=7 ttl=64 time=1.06 ms
64 bytes from 192.168.196.77: icmp_seq=8 ttl=64 time=0.959 ms
64 bytes from 192.168.196.77: icmp_seq=9 ttl=64 time=0.984 ms
64 bytes from 192.168.196.77: icmp_seq=10 ttl=64 time=0.872 ms
```

Figure 15. Ping Attack View

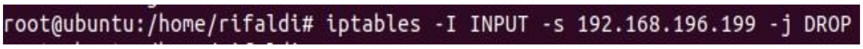
The test results with the ping command 192.168.196.77 from the client with IP address 192.168.196.199 will be detected directly on Snort. When Snort is run, the data packets will immediately appear and monitor the packet data sent or received through a particular network interface. If the network traffic is detected as a threat, Snort will display an alert message. Figures 14 and 15 are examples of Ping Attacks.



```
root@ubuntu:/home/rifaldi# snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=2440)
WARNING: No preprocessors configured for policy 0.
06/23-08:34:45.416619 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3133 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:7  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:45.416835 192.168.196.77 -> 192.168.196.199
ICMP TTL:64 TOS:0x0 ID:8042 IpLen:20 DgmLen:84
Type:0  Code:0  ID:4  Seq:7  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:46.421810 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3180 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:8  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:46.421901 192.168.196.77 -> 192.168.196.199
ICMP TTL:64 TOS:0x0 ID:8165 IpLen:20 DgmLen:84
Type:0  Code:0  ID:4  Seq:8  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:47.426531 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3406 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:9  ECHO
```

Figure 16. Snort Runs & Detects Intruders

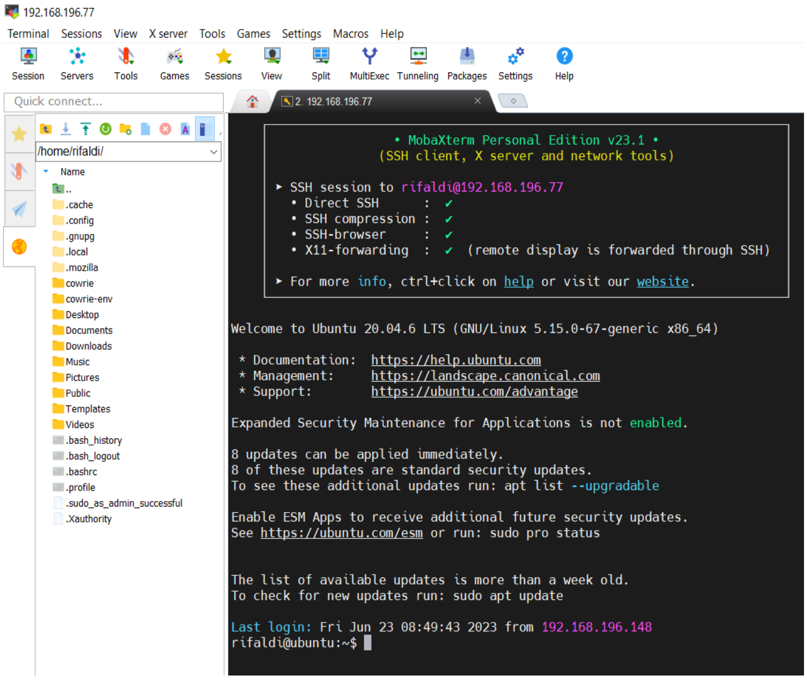
One way to overcome attacks on our network that have been detected by IPS Snort from denial of service (DOS) attacks in the form of ping of death carried out by IP clients into our server network is by blocking the client's IP address so that the client cannot access the server network. Snort Runs & Detects Intruders, shown in Figure 16. Meanwhile, the IPTables Commands to Prevent Attacks are shown in Figure 17.



```
root@ubuntu:/home/rifaldi# iptables -I INPUT -s 192.168.196.199 -j DROP
```

Figure 17. IPTables Commands to Prevent Attacks.

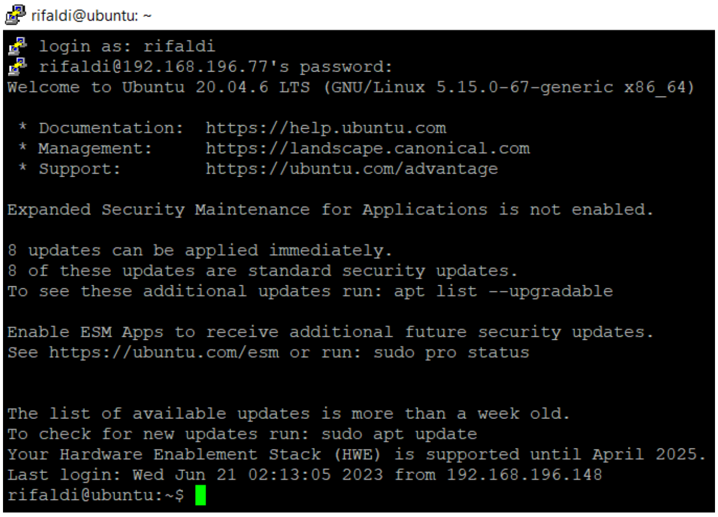
Display Figure 17 on how to block the IP address by using the iptables -I INPUT -s 192.168.196.199 -J DROP command on the Ubuntu Server operating system client, where drop means we drop or block the IP from the server. The Honeypot testing stage uses 2 applications, namely MobaXterm and PuTTY, which are open source applications. The PuTTY and MobaXterm applications found that the intruder was successfully redirected to the dummy server. Furthermore, in the Honeypot test using the MobaXterm application, it was found that the intruder was successfully diverted to the mock server. At this stage of testing, the intruder/intruder tries to remotely use port 22 with the username rifaldi and password rifaldi1601 on the Honeypot SSH. Like Figure 18.



```
          • MobaXterm Personal Edition v23.1 •
        (SSH client, X server and network tools)

 ➤ SSH session to rifaldi@192.168.196.77
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✓  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Jun 23 08:49:43 2023 from 192.168.196.148
rifaldi@ubuntu:~$ 
```

Figure 18. Intruder View successfully Redirected To Honeypot Server using MobaXterm

Furthermore, in testing Honeypot using the PuTTY application, it was found that the intruder was successfully diverted to the shadow server. At this stage of testing, the intruder/intruder tries to remotely use port 22 with the username rifaldi and password rifaldi1601 on the Honeypot SSH. Like Figure 19.

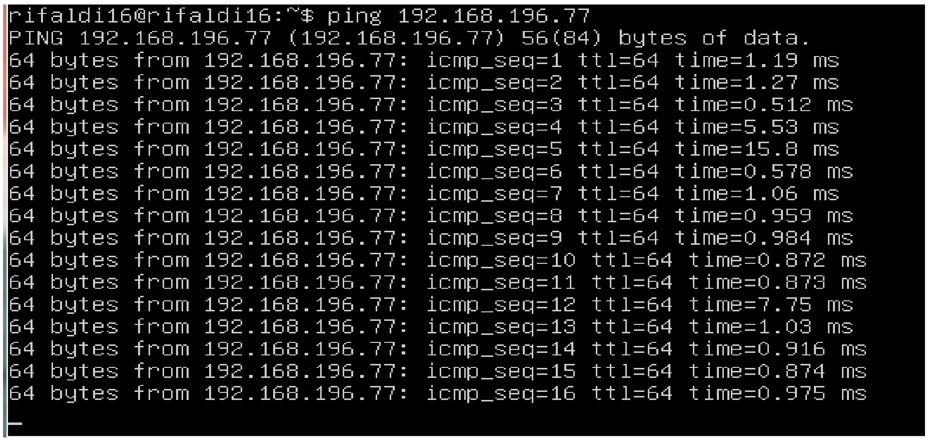


```
rifaldi@ubuntu: ~
login as: rifaldi
rifaldi@192.168.196.77's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Jun 21 02:13:05 2023 from 192.168.196.148
rifaldi@ubuntu:~$
```

Figure 19. Intruder View successfully redirected to Honeypot Server Using PuTTY

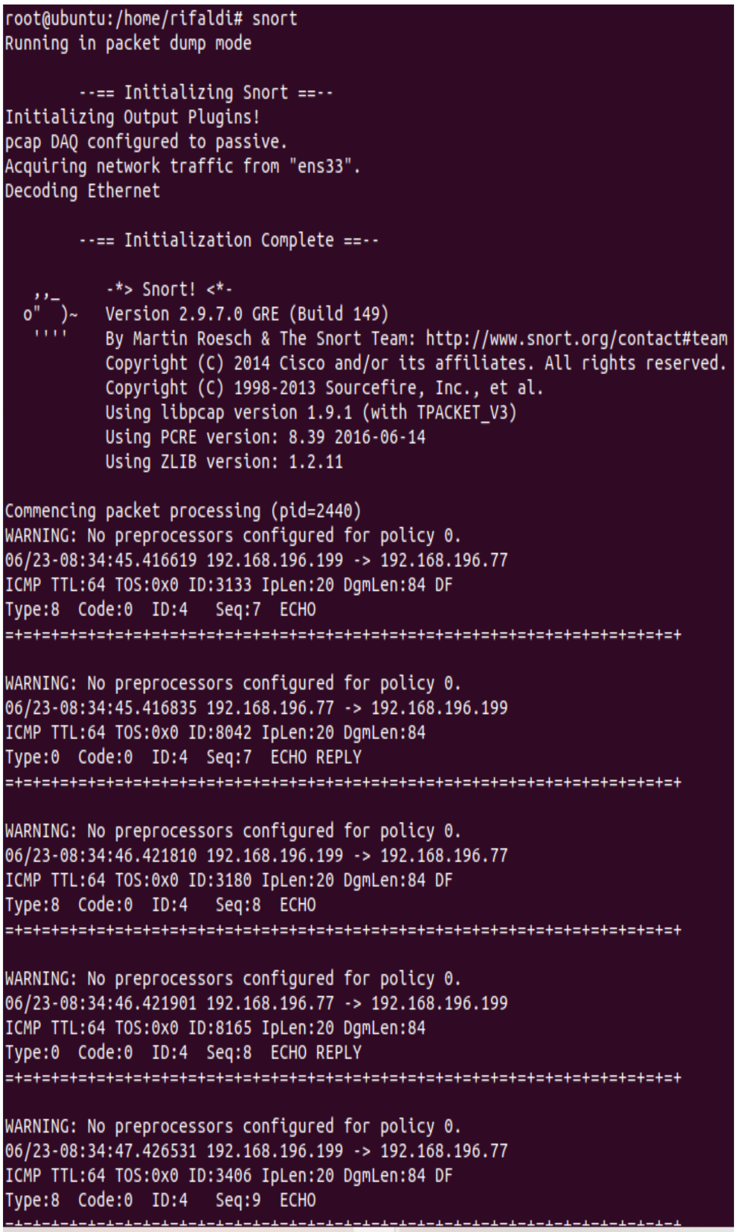
The test results aim to show whether the Intrusion Prevention System and Honeypot-based network security system that has been designed is running well or not, because the test results can provide information on the successful implementation of a network security system. Intrusion Prevention System (IPS) Testing Results: The test results with the ping command 192.168.196.77 from the client with IP address 192.168.196.199 will be detected directly by Snort. Snort is run, then the data packets will immediately appear, and monitor the data packets that are sent or received through a particular network interface. If network traffic is detected as a threat, Snort will display an alert message as shown in Figure 21.

Testing with IPTables aims to prevent attacks carried out by attackers. This research stage will be prevented by closing the port on the server so that the attacker cannot enter. This is used by the Linux Ubuntu operating system to prevent attacks on the server. The results of blocking the IP address in the test will try to carry out a denial of service (DOS) attack when the attacker's IP address has been dropped from our server. Result of Blocking IP Address shown in Figure 20.



```
rifaldi16@rifaldi16:~$ ping 192.168.196.77
PING 192.168.196.77 (192.168.196.77) 56(84) bytes of data.
64 bytes from 192.168.196.77: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.196.77: icmp_seq=2 ttl=64 time=1.27 ms
64 bytes from 192.168.196.77: icmp_seq=3 ttl=64 time=0.512 ms
64 bytes from 192.168.196.77: icmp_seq=4 ttl=64 time=5.53 ms
64 bytes from 192.168.196.77: icmp_seq=5 ttl=64 time=15.8 ms
64 bytes from 192.168.196.77: icmp_seq=6 ttl=64 time=0.578 ms
64 bytes from 192.168.196.77: icmp_seq=7 ttl=64 time=1.06 ms
64 bytes from 192.168.196.77: icmp_seq=8 ttl=64 time=0.959 ms
64 bytes from 192.168.196.77: icmp_seq=9 ttl=64 time=0.984 ms
64 bytes from 192.168.196.77: icmp_seq=10 ttl=64 time=0.872 ms
64 bytes from 192.168.196.77: icmp_seq=11 ttl=64 time=0.873 ms
64 bytes from 192.168.196.77: icmp_seq=12 ttl=64 time=7.75 ms
64 bytes from 192.168.196.77: icmp_seq=13 ttl=64 time=1.03 ms
64 bytes from 192.168.196.77: icmp_seq=14 ttl=64 time=0.916 ms
64 bytes from 192.168.196.77: icmp_seq=15 ttl=64 time=0.874 ms
64 bytes from 192.168.196.77: icmp_seq=16 ttl=64 time=0.975 ms
_
```

Figure 20. Result of Blocking IP Address



```
root@ubuntu:/home/rifaldi# snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=2440)
WARNING: No preprocessors configured for policy 0.
06/23-08:34:45.416619 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3133 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:7  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:45.416835 192.168.196.77 -> 192.168.196.199
ICMP TTL:64 TOS:0x0 ID:8042 IpLen:20 DgmLen:84
Type:0  Code:0  ID:4  Seq:7  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:46.421810 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3180 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:8  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:46.421901 192.168.196.77 -> 192.168.196.199
ICMP TTL:64 TOS:0x0 ID:8165 IpLen:20 DgmLen:84
Type:0  Code:0  ID:4  Seq:8  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/23-08:34:47.426531 192.168.196.199 -> 192.168.196.77
ICMP TTL:64 TOS:0x0 ID:3406 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:4   Seq:9  ECHO
```

Figure 21. Snort Detects Intruders

The Honeypot test results aim to show whether the security system that has been designed is running well or not, because the test results can provide information on the successful implementation of the server network security system, and the results of this test can compare server network security. The results of this test, intruders trying to remotely use port 22 with username rifaldi and password rifaldi1601, will be diverted directly by the Honeypot security system to a dummy server or redirected to Port 2222 on the SSH Honeypot. The results of the tests that have been carried out can be seen in the description of Table 1.

Table 1. Honeypot Testing Results

No	Application	IP Address Client	Port	Redirect to Port 2222	
				Successful	**Failed**
1.	MobaXterm	192.168.196.148	22	✓	-
2.	PuTTY	192.168.196.148	22	✓	-

Table 1 shows the results of Honeypot testing. In testing using 2 applications, namely MobaXterm and PuTTY, with the same IP Address, namely 192.168.196.148, and trying to remote using port 22 with username rifaldi and password rifaldi1601, it will be diverted directly by the Honeypot security system to a dummy server or redirected to Port 2222 on the Honeypot SSH.

5. Conclusions

Based on the discussion of network security based on Intrusion Prevention System (IPS) and Honeypot at the Regional Revenue Agency (Bapenda) of Padang City, conclusions can be drawn: [1] The results of research that has been done, Intrusion Prevention System (IPS) with Snort can detect networks that are illegal (dangerous for an operating system), prevent loss of data and information. Snort can find out what networks enter the operating system. [2] Server IPS can prevent Ping (ICMP) flood attacks carried out by attackers against the server by activating the firewall feature and configuring it with iptables. [3] The implemented honeypot successfully diverts port 22 SSH to port 2222, so that the attacker thinks port 22 is the actual SSH port. The limitations of this system contain the capabilities of the system that have been designed in the previous chapters. Here are some limitations of the system that has been designed: [1] The security system was created using VMware Workstation 17 Player, and the configuration of the security system on the network uses iptables on the Ubuntu operating system. [2] The security system implemented in the Intrusion Prevention System (IPS) network is only able to block attacks and detect attacks carried out by attackers on servers using the Linux Ubuntu Operating System. [3] The network security system built based on research conducted using Honeypot is only tested with MobaXterm and PuTTY.

Based on the above conclusions, suggestions that can be given include: [1] Further development should use more attack applications so that the rules of the IPS applied can work more optimally. [2] Aspects of server maintenance on a regular basis really need to be done so that the virtualized server can last a long time. Can develop software services that can provide applications and software that can be accessed via the internet and used simultaneously by all internet users, and add and implement other services.

Acknowledgments: The authors thank all those who have helped complete this research. Hopefully, this research can continue to be developed so that it can overcome cases of data theft via computer networks that can provide large losses if they are not resolved.

Author contributions: Rahmansyah, R., & Meiditra, I.: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft. **Rahmansyah, R., & Meiditra, I.**: Methodology, Software, Formal analysis, Data curation. **Rahmansyah, R., & Meiditra, I.**: Resources, Validation, Supervision. **Rahmansyah, R., & Meiditra, I.**: Conceptualization, Methodology, Software, Supervision, Project administration, Writing - review & editing.

Funding: The study was conducted without any financial support from external sources.

Availability of data and Materials: All data are available from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Additional Information: No Additional Information from the authors.

References

Abdulghani, A., Tarmin, T., & Solehudin, T. (2018). Sistem informasi pengelolaan administratif badan usaha milik desa (BUMDes) berbasis client-server studi kasus di Desa Sindangasih Kecamatan Karangtengah. Jurnal Ilmiah SANTIKA, 8(2), 241–254.

Adesty, I., et al. (2020). Penerapan Intrusion Prevention System (IPS) Suricata sebagai pengamanan dari serangan Distributed Denial of Service (DDoS). EasyChair Preprint, 2912.

Agni, I. H. (2019). Analisa DNS yang dimanfaatkan dalam filterisasi domain di jaringan WAN menggunakan open source. Jurnal IKRA-ITH Informatika, 3(88), 20–29.

Aini, N. (2019). Analisis jaringan local area network. Jurnal PROSISKO, 5(1). https://doi.org/10.31219/osf.io/htxwe

Alvian, M., et al. (2020). Investigasi serangan Backdoor Remote Access Trojan (RAT) terhadap smartphone. JURIKOM (Jurnal Riset Komputer), 7(4), 505–510. https://doi.org/10.30865/jurikom.v7i4.2301

Aminanto, A., & Sulistyo, W. (2020). Simulasi sistem keamanan jaringan komputer berbasis IPS Snort dan Honeypot Artilery. Aiti, 16(2), 135–150. https://doi.org/10.24246/aiti.v16i2.135-150

Arta, Y., et al. (2018). Simulasi implementasi Intrusion Prevention System (IPS) pada router Mikrotik. IT Journal Research and Development, 3(1), 104–114. https://doi.org/10.25299/itjrd.2018.vol3(1).1346

Astuti, I. K. (2018). Fakultas Komputer Indah Kusuma Astuti Section 01. Jaringan Komputer, 8.

Barends, J. K., et al. (2022). Perancangan dan analisis Intrusion Prevention System berbasis SNORT dan IPTABLES dengan integrasi Honeypot pada arsitektur Software Defined Network. Multinetics, 7(2). https://doi.org/10.32722/multinetics.v7i2

Bastian, A., et al. (2017). Rancang bangun sistem informasi manajemen peternak ayam pada Koperasi Sinar Mulya menggunakan Microsoft Visual Basic 2010 .Net. Studia Informatika: Jurnal Sistem Informasi, 10(2), 135–143.

Bossuyt, P., et al. (2018). Defining endoscopic remission in ileocolonic Crohn's disease: Let's start from scratch. Journal of Crohn's and Colitis, 12(10), 1245–1248. Retrieved October 31, 2022, from http://doditsuprianto.blogspot.com/p/keamanan-jaringan-dengan-pengertian-ips.html

Cahyanto, T. A., et al. (2017). Analisis dan deteksi malware menggunakan metode malware analisis dinamis dan malware analisis statis. Justindo: Jurnal Sistem & Teknologi Informasi Indonesia, 2(1), 19–30.

Dasmen, R. N., et al. (2022). Penerapan Snort sebagai sistem pendeteksi serangan keamanan jaringan. Jurasik: Jurnal Riset Sistem Informasi dan Teknik Informatika, 7(1), 8. https://doi.org/10.30645/jurasik.v7i1.409

Fachri, B., & Harahap, F. H. (2020). Simulasi penggunaan Intrusion Detection System (IDS) sebagai keamanan jaringan dan komputer. Jurnal Media Informatika Budidarma, 4(2), 413. https://doi.org/10.30865/mib.v4i2.2037

Fitriana, N., & Khasanah, F. N. (2018). Honeypot menggunakan Honeyd sebagai solusi keamanan jaringan dari aktivitas serangan. Bina Insani ICT Journal, 5(2), 143–152.

Haryani. (2017). Sosialisasi internet sehat sebagai upaya pornografi di internet bagi pemuda pemudi Gedongkuning, Tegaltandan, Banguntapan, Bantul, 25–28.

Hawari, M. S. (2017). Penerapan IPTables firewall pada Linux dengan menggunakan Fedora. Jurnal Manajemen Informatika, 6, 198–207.

Sobirin, I., Putra, F., & Putra, R. D. (2020). Sistem keamanan jaringan berbasis Intrusion Prevention System dan Honeypot pada PT Matra Agung Persada.

Irawan, D. (2017). Blokir malware berbahaya melewati proxy menggunakan router Pfsense dan paket HAVP. Jurnal Manajemen Informatika, 7(2), 53.

Laksana, I., & Rosyid, N. R. (2017). Implementasi Honeypot sebagai pemantau parameter pada HTTP request untuk mengetahui tujuan serangan. Retrieved October 30, 2022, from http://etd.repository.ugm.ac.id/penelitian/detail/115923

Maharani, D., et al. (2021). Penyuluhan manfaat menggunakan internet dan website pada masa pandemi COVID-19. Abdiformatika: Jurnal Pengabdian Masyarakat Informatika, 1(1), 1–7. https://doi.org/10.25008/abdiformatika.v1i1.130

Mufida, E., et al. (2019). Rancang bangun sistem informasi inventory pada salon kecantikan. Jurnal Mantik Penusa, 3(3), 99–102.

Muftikhali, Q. E., et al. (2018). Optimasi algoritma genetika dalam menentukan rute optimal topologi cincin pada Wide Area Network. Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer, 13(1), 43. https://doi.org/10.30872/jim.v13i1.1007

Muhaimi, A., et al. (2019). Analisa penerapan Intrusion Prevention System (IPS) berbasis Snort sebagai pengaman server internet yang terintegrasi dengan Telegram. Jurnal Bumigora Information Technology (BITe), 1(2), 167–176. https://doi.org/10.30812/bite.v1i2.611

Munawar, Z., et al. (2020). Keamanan jaringan komputer pada era Big Data. Jurnal Sistem Informasi J-SIKA, 2, 14–20.

Muqorobin, M., et al. (2019). Implementasi Network Intrusion Detection System (NIDS) dalam sistem keamanan Open Cloud Computing. Majalah Ilmiah Bahari Jogja, 17(2), 1–9. https://doi.org/10.33489/mibj.v17i2.205

Muslihah, I., & Rozaq, N. A. (2021). Sistem informasi akuntansi keuangan sekolah berbasis client-server. Jurnal Data Mining dan Sistem Informasi, 2(2), 45. https://doi.org/10.33365/jdmsi.v2i2.1343

Noor, E., & Chandra, J. C. (2020). Implementasi firewall pada SMP Yadika 5 Jakarta. IDEALIS: Indonesia Journal Information System, 3(1), 449–456. https://doi.org/10.36080/idealis.v3i1.2088

Novrianda, R. (2018). Implementasi metode VLSM (Variable Length Subnet Mask) pada pemetaan IP address LAN (Local Area Network) STIPER Sriwigama Palembang. Computatio: Journal of Computer Science and Information Systems, 2(2), 112–118.

Nurlelah, F., & Yohana. (2022). Perancangan sistem informasi pelayanan administrasi terpadu satu pintu berbasis web. Jurnal Teknik Komputer AMIK BSI, 9(2), 16–25. https://doi.org/10.31294/jtk.v4i2

Pamungkas, P. D. A. (2018). Analisis cara kerja sistem infeksi virus komputer. Bina Insani ICT Journal, 1(1), 15–40.

Pangestu, P., & Desmira. (2021). Analisis optimalisasi kinerja jaringan MAN pada layanan internet berbasis Mikrotik di PT. Bina Technindo Solution. Jurnal PROSISKO, 8(1), 8–17.

Pratama, R. D., et al. (2020). Perancangan dan implementasi Wide Area Network menggunakan Q-in-Q tunneling pada Telkom School. E-Proceedings of Engineering, 7(2), 4841–4856.

Razak, R. (2018). Pendeteksian dan pencegahan serangan pada jaringan menggunakan Snort pada Linux Ubuntu. Journal of Controlled Release, 11(2), 430–439.

Riska, P., et al. (2018). Sistem keamanan jaringan komputer dan data dengan menggunakan metode Port Knocking. Jurnal Sistem Informasi dan Komputer, 1(2), 53–64.

Rismawati, N., & Mulya, M. F. (2020). Analisis dan perancangan simulasi jaringan MAN (Metropolitan Area Network) dengan dynamic routing EIGRP dan algoritma DUAL menggunakan Cisco Packet Tracer. Jurnal SISKOM-KB, 3(2), 55–62. https://doi.org/10.47970/siskom-kb.v3i2.147

Ronaldo, C., & Chandra, W. (2020). Anti-WebShell PHP backdoor scanner pada Linux server. ILKOM Jurnal Ilmiah, 12(2), 143–153.

Ryansyah, M., & Maulana, M. S. (2018). Malware security menggunakan filtering firewall dengan metode Port Blocking pada Mikrotik RB 1100AHx2. Sistem dan Teknologi Informasi, 6(3), 6–10.

Sampurno, D. S., et al. (2019). Implementasi pembuatan distro Linux untuk keperluan laboratorium informatika. Jurnal Infra Petra, 2(1), 2.

Santoso, J. D. (2019). Keamanan jaringan nirkabel menggunakan wireless intrusion. INFOS Journal, 1(3).

Sari, W. P., & Putra, I. N. A. P. (2017). Analisis serangan hacker menggunakan Honeypot High Interaction (HIHAT). Jurnal TIARSIE, 14(1), 13–18.

Sasmita, R. S. (2020). Pemanfaatan internet sebagai sumber belajar. Jurnal Pendidikan dan Konseling, 1, 1–5.

Simanjuntak, P., et al. (2018). Analisis penggunaan jaringan LAN pada PT USDA Seroja. CBIS Journal, 6(1), 23–28.

Suhendi, H., & Cahyo, W. D. (2021). Jaringan menggunakan Snort sebagai Intrusion Prevention System (IPS) pada jaringan internet. Jurnal Sistem Informasi dan Komputer, 3(2), 60–68.

Supendar, H., & Handrianto, Y. (2017). Teknik Frame Relay dalam membangun Wide Area Network dengan metode Network Development Life Cycle. Bina Insani ICT Journal, 4(2), 121–130.

Susanti, R. E., et al. (2022). Implementasi Intrusion Prevention System (IPS) OSSEC dan Honeypot Cowrie. Jurnal Sisfokom (Sistem Informasi dan Komputer), 11(1), 73–78. https://doi.org/10.32736/sisfokom.v11i1.1246

Suwanto, R., et al. (2019). Implementasi Intrusion Prevention System (IPS) menggunakan Snort dan IPTable pada monitoring jaringan lokal berbasis website. Jurnal Komputer dan Aplikasi, 7(1), 97–107.

Tangkowit, A. E., et al. (2021). Analisis dan perancangan jaringan komputer di Sekolah Menengah Pertama. Edutik: Jurnal Pendidikan Teknologi Informasi dan Komunikasi, 1(1), 69–82. https://doi.org/10.53682/edutik.v1i1.1044

Tujni, B., & Alfiansyah, A. H. (2018). Perancangan pemetaan IP address menggunakan metode VLSM di PT KAI Divre III Palembang Sumatera Selatan. Prosiding Semhavok, 40–47.

Utomo, Y. A., et al. (2018). Membangun sistem analisis malware pada aplikasi Android dengan metode reverse engineering menggunakan REMnux. Jurnal Manajemen Informatika, 4(3), 2000–2012.

Wahyudi, F., & Utomo, L. T. (2021). Perancangan security network Intrusion Prevention System pada PDTI Universitas Islam Raden Rahmat Malang. Edumatic: Jurnal Pendidikan Informatika, 5(1), 60–69. https://doi.org/10.29408/edumatic.v5i1.3278

Wibawa, G. H. P., et al. (2020). Analisis data log Honeypot menggunakan metode K-Means Clustering. Jurnal Ilmiah Merpati, 8(1), 13. https://doi.org/10.24843/jim.2020.v08.i01.p02

Pradipta, Y. W. (2017). Implementasi Intrusion Prevention System (IPS) menggunakan Snort dan IP Tables berbasis Linux. Manajemen Informatika, 7, 21–28.