Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Honeypot for Securing Networking Devices

Ms. Bhakyalashmi E.[1], Mr. Gokulkrishnan G.[2], Dr. S. Uma [3]

[1,2] *II M.Sc Computer Science, Dr. N.G.P. Arts and Science College,Coimbatore-48, Tamil Nadu, India.*

[3]*Associate Professor, Department of Computer Science, Dr. N.G.P. Arts and Science College, Coimbatore-48, Tamil Nadu, India.*

Abstract: Abstract: Today's society needs security because of the increasing risks and cybercrimes. The security and stability of our home networks must be given top priority in the ever changing world of the internet today. The Internet of Things (IoT) has transformed simple networks into increasingly intricate systems, raising the possibility of threats and vulnerabilities. In order to efficiently filter out harmful traffic and keep an eye out for threats, this paper proposes a comprehensive defense system that consists of an Intrusion Detection System (IDS), a decoy honeypot server, and a packet analyzer.

Keywords: *Internet of Things, honeypot, home network, Raspberry Pi security, Intrusion Detection System*

I. Introduction

Security is a recurring issue in this dynamic world of cybercrimes, growing hazards, and ongoing progress. To strengthen security, reduce susceptibility, maintain integrity, and prevent manipulation, our initial objective is to safeguard and defend our Internet of Things devices and home network from cyber-attacks, as this is an essential part of information security. Throughout the entire paper, intrusion monitoring software and deception technologies are used in conjunction with a honeypot module to monitor and secure the network. In conjunction with the Raspberry Pi, a decoction-focused development, honeypot lowers the cost and simplifies the implementation of our modest network defense system. If built properly, intrusion detection systems and honeypots ought to function effectively.

Honeypots imitate systems that hackers want to attack in an effort to attempt to deflect attacks and identify malicious intent by collecting information about the assault and the victim. Program for monitoring intrusions keeps track of internet activity and sends out alerts when certain suspicious traffic packets are found. The network log data transmission is visible in real-time using the packet analyzer. Both tiny home networks and small business networks may benefit from this and other easy-to-install and low-cost maintenance small home network safety strategies. All log data is saved on this server installation for help in analyzing attacker behavior in the future.

This endeavor uses a variety of open-source software modules for network administration and analysis on the third-generation Raspberry Pi. This stream of unfiltered, real-time traffic is monitored and analyzed by the sensor module of the IDS that we have set up. T-Shark when positioned like a Raspberry Pi packet analyzer can perform more complex packet sniffing and analysis than standard network protocol analyzers, which just show users a simple display of network packets. In the ever-changing era of quick data creation, inexpensive internet connection, and global data sharing, home network security has been emphasized because neglecting to do so exposes us to Honeypot, and packet analyzer may significantly improve the security of a small or home network [1].

II. Background

A. Raspberry Pi

The Raspberry Pi, invented by Ebben Upton, is a single-board computer in the most basic sense imaginable. One of the company's founders, Ebben The creation of the first Raspberry Pi by Upton was aimed at addressing the issue of declining enrollment in computer science at universities. It was created to inspire students to seek fundamental education in computer technology in colleges and institutions and provide the general public with a

low-cost tool that may enhance young people's hardware and programming skills. The Raspberry Pi is a tiny, affordable, Linux- based device that can allow all program to operate at a reduced level of power consumption even if its computational capacity is plainly much lower than that of a regular computer. Figure 1 depicts the Raspberry Pi 4's architecture. A moderately priced, tiny mini- computer with all the capabilities of a desktop or laptop that can be operated with a conventional keyboard, mouse and hooked into an output display. It is a versatile and portable tool that can do all task that you would want to perform on a regular computer, including web surfing, challenging video game play, and developing cost-effective and energy-efficient home automation systems. Almost every industry uses this technology regularly for a variety of tasks, and it often has the ability to interact with the outside world.

Figure 1: Raspberry Pi 4 Model B Board

B. Intrusion Detection System (IDS)

An Intrusion Detection System is a piece of hardware or software that records incoming and outgoing network traffic in accordance with specified criteria and analyses and stores data. When it notices any unusual activity, it will notify us. Intrusion detection systems are used in networks to offer comprehensive network protection by spotting and preventing threats while also gathering information for further analysis[2].

An IDS can be categorized according to types:

1) Network-Based IDS: This IDS can monitor all network device incoming and outbound traffic since it is installed on a physical network.

2) Host Based IDS: To enhance network security, a Host-based Intrusion Detection System (HIDS) can be installed on one or multiple computers under the management of a registered network administrator. By monitoring incoming and outgoing system packets, the HIDS alerts the owner upon detecting any suspicious or harmful activity. The system leverages up-to-date device archives and merges them with previously collected data for a more comprehensive view of potential threats.

An IDS may be categorized according on the detection technique as follows:

1) Signature-Based detection: Also referred to as rule-based or signature-based detection, this sort of detection maintains track of all the data flowing through their infrastructure and compares it to a number of databases that include event signatures for detecting threats. This method takes advantage of regulations or paperwork produced by or held by security experts.

2) Anomaly-Based Detection: This kind of detection keeps track of user, server, and network activity over time and scans it for anomalies. When such behavior is discovered, alerts are then produced. All of this relies on rules that were developed based on actual assaults [3].

C. Honeypot

On the network, a honeypot would be set up to act as a lure for attackers and collect data on the intrusion and the attack. A honeypot crudely replicates networks that might be the target of hackers. The primary goals of deploying

a honeypot are to deter attacks and gather data that can be used for deeper threat analysis. The security element of the honeypot that lures burglars in by simulating a labyrinth. A deliberately infected operating program allows an attacker to discover defects, which researchers may examine to strengthen security precautions. Any computing resource can use a honeypot by applying it to file servers, routers, applications, and networks. They may keep a detailed record of the assault utilizing equipment that is automated for recording and tracking. The IP address was generated by a honeypot, and the administrator is always keeping an eye on it. The honeypot must wait before engaging with the system if any interaction is detected as suspicious. If not completely redirect the intrusion, they may very well waste the intruder's time and, when used properly, can be very effective.

Honeypots can also be paired with software to generate alerts and log data. Honeypots can be classified as producing honeypots or research honeypots, depending on their intended use. Overall, honeypots serve as an effective means of maintaining network security and identifying potential threats [4].

D. Packet Analyzer

A network packet analyzer captures, analyzes, and decodes real-time network traffic to help identify potential network problems. By tracking and presenting both incoming and outgoing network traffic in a user-friendly manner, these tools allow users to effectively monitor their network. Packet captures can be run on dedicated workstations or specialized hardware connected to a network, and can be used as a part of an overall application security program.

Monitoring systems are capable of controlling several networks. Firewalls and anti-malware software are examples of equipment that is crucial to network management and security. However, there are legitimate and illegitimate uses for analysts. Sniffers are considered illegal when they are used by hackers to obtain access without authorization in order to monitor or steal private information. However, network protection analyzers that are installed with the owner's consent to provide network control and troubleshooting can be regarded as legitimate sniffers. Packet analyzers that capture network data passively are often not detected, despite the fact that the active analyzers are likely to be able to identify them.

Packet analyzers collect and keep track of the network data that is available on the network interface to which they are deployed. How simple it is to collect the data will depend on how the network analyses are configured. The design of the network and the configuration of the network switches in a wired network define the information that the analyses collect. Analysts may only record data from one channel at a time in a wireless network unless multichannel recording is allowed on multiple network interfaces. When the information is gathered, it will be provided in a format that people can use to analyses it. Data is used to assess weaknesses and defects, such as locating network requests that go unrecognized [5].

Iii. Proposed System

The suggested system has a straightforward hardware and software setup. We're using a Raspberry Pi 4B module as our hardware, which we can physically or wirelessly connect to a Wi-Fi network. First, we will configure the hardware environment as part of the preliminary process. A memory card booted with the operating system, an external mouse and keyboard, a display monitor, a Raspberry Pi module with an HDMI connection, and all of these items are required. As a result, a mouse and keyboard are now attached to USB ports for input and the Raspberry Pi module is connected to an HDMI interface for output. Through the use of a micro-USB cord, the device will be controlled. To set up the Raspberry Pi for our purposes, we have loaded a Raspbian-Stretch-2018-Desktop-Image onto its memory card. Once the operating system is installed, it is necessary to connect the module to the internet and update and upgrade the Raspbian OS packages.

Figure 2 shows the installation of an intrusion prevention system, a packet analyses, and a honeypot application between the router access point and the local network. We also built a network-based intrusion detection system, a decoy honeypot, and a packet sniffer-analyzer. A number of clients are connected to the server, which is configured to accept all incoming messages and store them in the information, as a result of the centralization of the data that has been received.

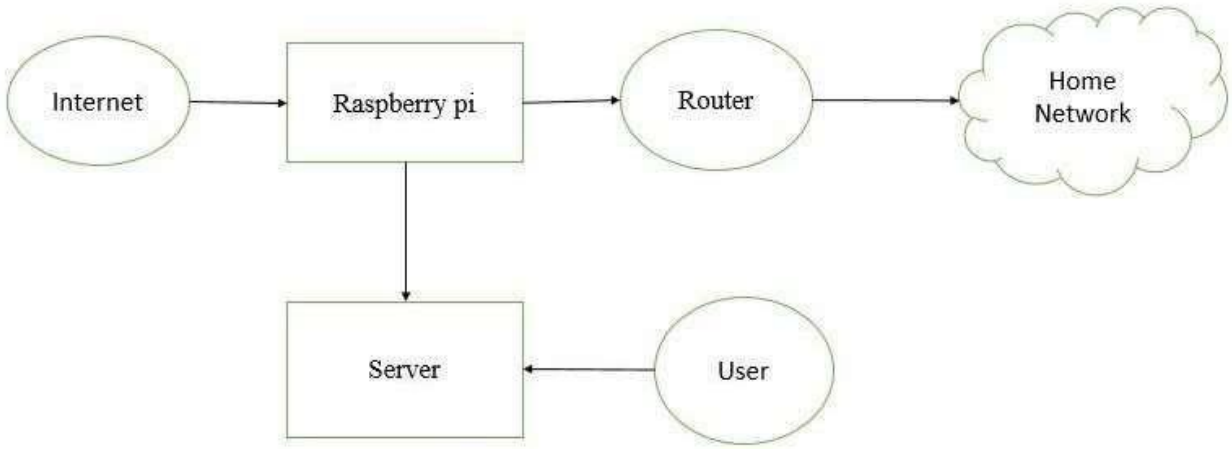

Figure 2: Architecture of the system

Raspberry Pi was provided the DHCP configuration for the eth0 port and was immediately granted permission to use the eth0 interface if we are using Ethernet over LAN cable. If we are utilizing LAN cable as Ethernet, then Raspberry Pi was given the DHCP setup for the eth0 interface, and the eth0 interface was permitted upon start.

A. Honeypot

In addition to SSH and brute force attacks, we are configuring Cowrie, a medium-level SSH honeypot interface, with our Raspberry Pi honeypot to watch full shell interfaces for threats.

1) Cowrie: The Python-based medium-level interface honeypot was created to monitor all shell activity during an attack, including the recording of brute force cracking attempts. Criminals utilize it as a laborious, annoying trial-and-error technique to get around passwords and paraphrases. Before continuing to enter into the network and tricking themselves into believing they are accessing the device's real SSH session, this honeypot helps any intruder establish contact. When the brute-force attempt to crack the authentication is successful, the attacker is pointed towards the bait interface they would use for communication. If the intruder is successful in accessing this bait network, it may track and record any disruptive behavior the intruder is involved in while simulating the real device. Cowrie provides fictitious file systems that make it possible to input and remove data efficiently. In this approach, the majority of the data accessed during the connection is still saved.

2) Domine: It's a honeypot run on Python with Low-level Interaction. Domine's objective is to stop malware by exploiting vulnerabilities found by networked services. The honeypot's main objective is to get a copy of the sample malware. Honeypot Domine has created an environment to detect and collect malware assaults so that malware may be recognized before it becomes a severe issue in the modern information technology world. By disclosing the virus to outside monitoring service providers, Honeypot Domine has been installed on environment to detect and gather malware assaults. All honeypot logs and malware samples that were discovered and detected by the raspberry pi module will be kept in a database.

B. Packet Analyzer

Wireshark's command-line version for packet capture and analysis, T-shark, may be installed to allow us to create a network analyses and sniffer on our Raspberry Pi.

T-shark: T-shark is a network administration and reporting program that runs on terminals. This is a network analysis tool that monitors and evaluates traffic across linked networks. As a result, anybody may now easily analyse packets using later- gathered network logs and have real-time access to network information. If neither option is provided, T-shark should behave like TCP-dump. It is feasible to employ packet sniffers as a vital network security and monitoring tool. These were at first used to track how much infrastructure was being used, spot network overloads, and pinpoint inefficiencies. They might really be used to classify network vulnerabilities and pinpoint susceptibility weak areas. When a network failure occurs, for example, they could be used to pinpoint

the main reason for an issue. T-shark records packets that essentially contain information about the packet, the layer, the amount of free memory, and a graphical explanation of each of the aforementioned details [6].

IV. Implementation

The honeypot is designed using the Raspberry Pi hardware module, which is a cost-effective alternative to using a modern system in a small infrastructure. The data collected and logs are stored for further analysis on local or cloud servers. The honeypot is intended to simulate a fake network that attracts malicious users to gather information about the invader and target before securing the network. To achieve this, we installed the Intrusion Detection System software on a Raspberry Pi module from the official source. Snort is an open-source tool used for monitoring and alerting the user and administrator. After installation, we modified the snort.config file to apply changes to it. Users can either create their own rule set from scratch or obtain pre-existing rule sets from sources. Barnyard2 is installed as a Snort output component to process warnings generated by Snort and output them as a human-readable database layout that can be customized. The deployment of Pulled-Pork, a PERL-based rule creation method for intrusion detection programs such as Snort and Suricata, is then executed. The relevant set of rules will be automatically created when configured to recognize the Snort variant. The pulled-pork provided by Snort sources can be helpful if the user is unable to manually configure the rule set. To avoid manually starting Snort, Barnyard2, or Pulled-Pork every time, a small start- up shell script can be created that launches automatically when the Raspberry Pi boots up. All honeypot logs are stored in the directory /var/log/IDS name, and the services running in the module can be used to check whether IDS is operational [7].

Next, we plan to install a medium-level interaction honeypot called Cowrie on the Raspbian operating system. Cowrie, a Python-based program, can effectively monitor all shell activity during a bulk operation and record brute force cracking attempts. Prior to installation, we will review the installation requirements and then proceed with downloading and extracting the Cowrie deployment file. After creating a shell script and checking its status, we will install and activate the honeypot. In the event of suspicious traffic, the user will receive a notification via email or alert. Additionally, we will install Domine, another honeypot on the Raspberry Pi module, designed to detect and remove malware from the network. It maintains a database of every malware, logs its activity, and can be shared for global examination on a cloud server. The logs are stored at /var/log/Domine.

Thirdly, we'll put T-shark, a network monitoring and assessment framework built on terminals that monitors real-time data flow, into action. The T-shark software will be obtained from an official site, installed in the Raspberry Pi module, and then used to monitor all network traffic in real time, generate logs, and alert the user.

Last but not least, we have two choices for where to save logs: locally or on a server. The /var/logs/ subdirectory will house the logs that are generated by each IDS and honeypot. For the server, we need to construct a server that runs the System on top of Ubuntu OS. Here, we'll be using the MHN server, an Ubuntu OS setup that will store all the logs that the Raspberry Pi has gathered and present them in an understandable manner. The MHN server has to be downloaded, installed on Ubuntu OS, and set up to operate either as a virtual machine or in the cloud. Here, the server is configured in a virtual machine, and we are using a browser to access it.

V. Results And Discussions

A. To conduct a thorough test of the IDS module, Packet Analyzer module, and honeypot program we provided, it was necessary to verify that each module was functioning correctly.

B. Output of Intrusion Detection System

The attacking machine checked the Raspberry Pi for any registered alerts or data gathered by Snort to determine if it had been detected. Figure 3 shows the ping command being executed by the attacker’s machine.

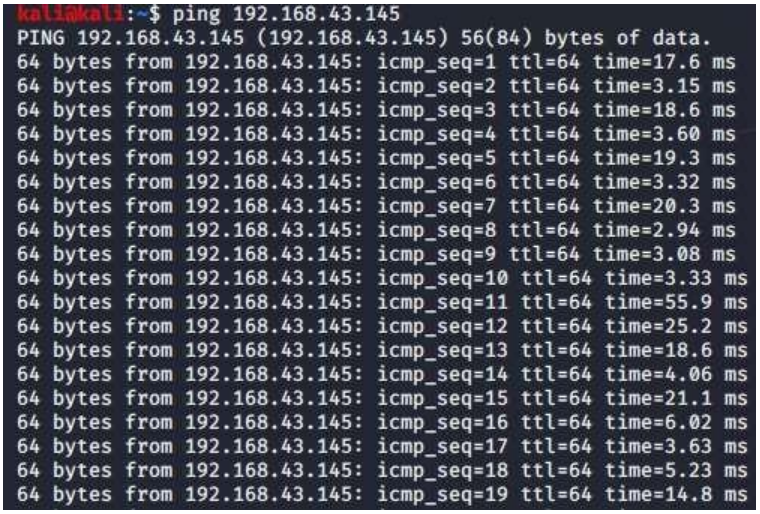

Figure 3: Ping performed by the attacker system

Figure 4 illustrates how the Raspberry Pi interface has been configured with the ICMP specifications. The attacker machine's ICMP ping queries were successfully recorded by Snort Intrusion Detection System.

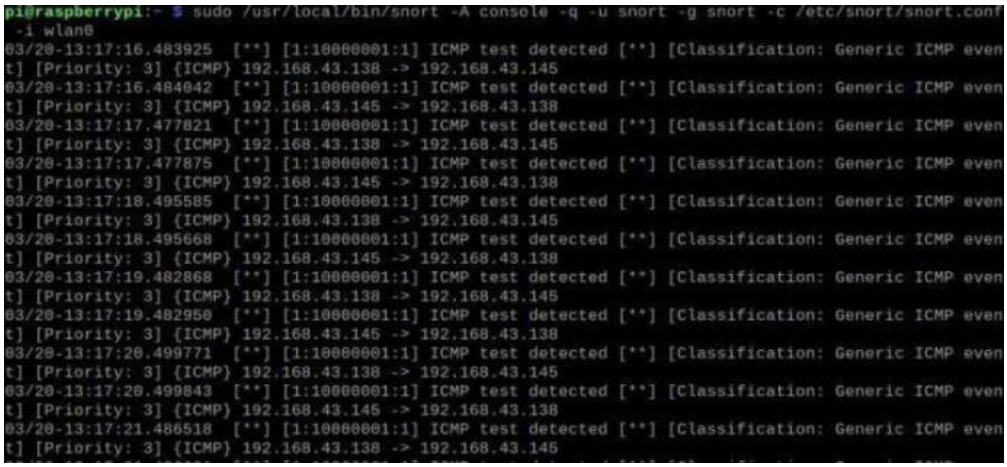

Figure 4: Display the capturing of packets by IDS

C. Output of Honeypot Analysis

We'll use Nmap to scan Cowrie and then use the Kali- Linux systems to launch a brute force attack. As seen in Figure 5, the target device is scanned for open- port vulnerabilities.

Figure 5: Display weakness capture by scanner

Now, while using the brute-password, we will attempt to use SSH terminal to access any open ports. In order to gather sensitive client information and usernames and passwords that are often used for unauthorized access, such those shown in Figure 6, the brute force attack uses a trial-and-error methodology.

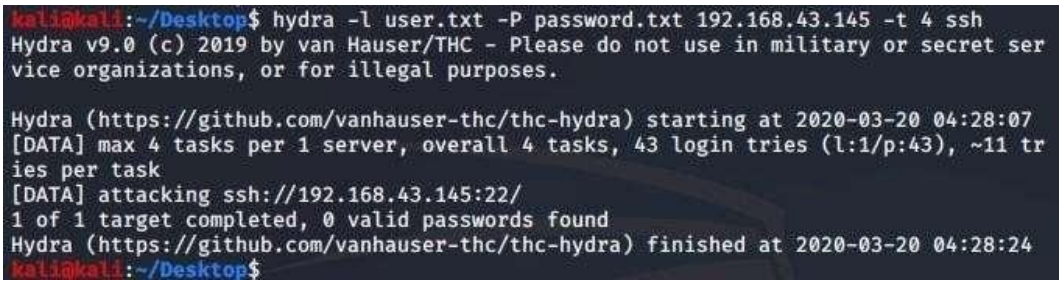

Figure 6: Brute force attack attempt

D. Output of MHN Server

All of the IDS and honeypot logs will be received by the MHN Server, which will then use them to visualize the data in graphs that even non-technical users can readily understand.

In order to assist people better understand network assaults, all of the logs gathered by the honeypot, IDS, and T-shark were shown in a form that is simple to understand by humans in Figures 7 and 8.

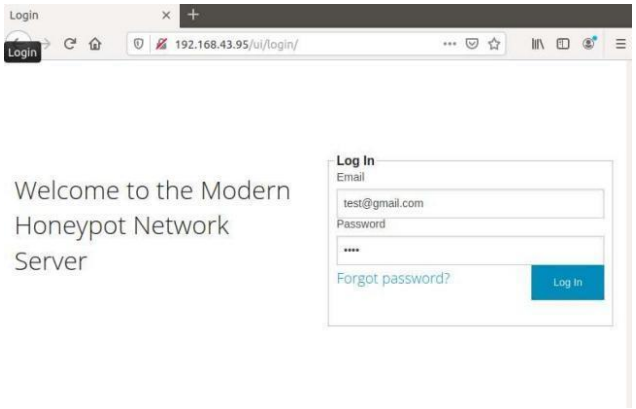

Figure 7: Data logging on the server

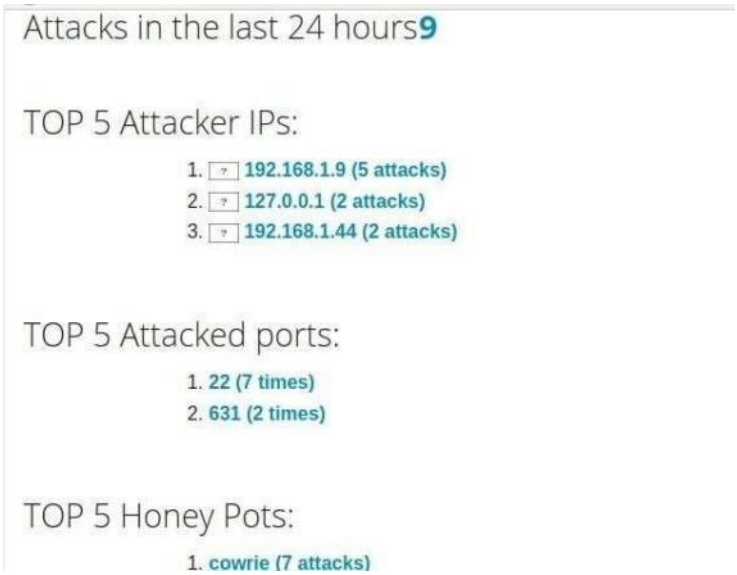

Figure 8: Information gathered

VI. Conclusion

Security is without a doubt one of the largest issues confronting modern society. Utilizing technologies such as Intrusion Detection Systems, honeypots, and packet analyzers can effectively combat security threats, safeguard confidential information, and ensure data privacy. The Raspberry Pi surveillance system can also benefit from these technologies.

The cost of defending against a small network is cheaper than the cost of setting up a monitoring system on a single machine. In this research, we suggest using a Raspberry Pi 4 with an IDS and a honeypot to build devices that would monitor the small network and act as a detection alarm system. Packet Analyzer functions as a T-shark, Cowrie and Domine as honeypots, and Snort as an IDS. The deployment and operation of a honeypot using a Raspberry Pi 4 board was cost-effective and efficient. As we move forward, instead of installing local servers, we can utilize cloud servers that can be monitored remotely.

VII. References

[1] T. Zitta, M. Neruda, and L. Vojtech, “The security of RFID readers with IDS/IPS solution using Raspberry Pi,” 2017 18th International Carpathian Control Conference (ICCC), 2017.

[2] A. K. Kyaw, Y. Chen, and J. Joseph, “Pi-IDS: evaluation of open- source intrusion detection systems on Raspberry Pi 2,” 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec), 2015.

[3] S. Mahajan, A.M Adagale and C. Sahare, “Intrusion Detection System Using Raspberry Pi Honeypot in Network Security,” 2016 International Journal of Engineering Science and Computing Volume 6, Issue 3, 2016.

[4] J. Jeremiah, "Intrusion Detection System to Enhance Network Security Using Raspberry PI Honeypot in Kali Linux," 2019 International Conference on Cybersecurity (ICoCSec), Negeri Sembilan, Malaysia, 2019, pp. 91-95.

[5] M. Coşar and H. E. Kiran, "Measurement of Raspberry Pi performance in network traffic analysis," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4.

[6] Y. Turk, O. Demir, and S. Gören, “Real Time Wireless Packet Monitoring with Raspberry Pi Sniffer,” Information Sciences and Systems 2014, pp. 185– 192, 2014.

[7] C. Bousaba, T. Kazar, and W. Pizio, “Wireless Network Security Using Raspberry Pi,” 2016 ASEE Annual Conference & Exposition Proceedings, 2016.