International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-7, May 2020

Securing a Small Network by using Raspberry Pi Honeypot

Monica Catherine, Ashok Kumawat

Abstract: ***In this ever changing world, with cybercrimes and increasing threats, security is becoming constantly an concern. As security became a basic requirement and a need in the times of today with some sort of access to the data and interaction on the internet it is increasingly unstable. This is incredibly necessary to maintain our home network up-to- date. In terms of reliability, reduce vulnerability and maintain integrity and untampered. As a basic security characteristic of information security, it is our top priority that we defend and safeguard our Internet of Things devices and home network from cyber threats. Honeypot a development focused on decoction together with raspberry pi makes our small cost effective and quick to implement network defense security. The whole paper deals with implementing a Honeypot deployed in module Raspberry Pi operating Intrusion Monitoring Program and deception technology as a induce for monitoring and securing the network. This should prove successful intrusion detection systems and honeypots solutions if implemented correctly.***

Keywords: ***Intrusion Detection System, Internet of Things, Honeypot, Network, Raspberry Pi Security***

I. INTRODUCTION

Throughout today's environment, with cybercrimes and increasing threats, security is both a necessary need and a necessity. When every form of internet intercommunication and data storage is becoming unstable, maintaining our well-protected small home network is becoming increasingly important in terms of security, the closure of vulnerabilities and the protection of data transparency and moral integrity increased essentially. Moreover, as IOT or the Internet of Things grew, their simple basic home networks slowly developed towards sophisticated advanced networks that rendered themselves increasingly vulnerable towards threats. So here paper has been focused on developing a defense all together with other as one Security project where it server to protect the small infrastructure through serving like an IDS as well as filtering harmful packets, and also functioning such as decoy honeypot server as well as packet analyzer for monitoring and evaluate threats. These compact form factor, cheap expense, and respectable processing capacity of the Raspberry Pi are the reasons it was selected as the target device.

Revised Manuscript Received on May 30, 2020.
***** Correspondence Author

Monica Catherine*, Assistant Professor, Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
Ashok Kumawat, PG Scholars , Department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

Intrusion monitoring program tracks internet usage and produces warnings when some suspicious traffic packets as detected and honeypots mimic systems which hackers intend for target just as view to trying to divert attacks and detecting malicious intent through catching information of attack and the target. The packet analyzer provides a real-time viewing the log data flowing to through the network. Small home network protection strategies, such as this, simple installation but also low cost maintenance, can help to improve protection of either small home networks or small businesses. With this installation of the server where all the log data is store for the future investigation and analysing the attacker behavior. This project uses numerous open source software modules for network management and review, built as a third generation Raspberry pi. We use Snort as to setup their IDS for Instance, where the sensor module, Snort, analyzes raw live traffic and monitor this streams. Tshark is designed to setup as packet analyzer on the raspberry pi, whereas commonly deployed network protocol analyzer competent for sniffing and viewing network packets for user as simple format. Finally, Cowrie and Dionaea is designed to be the honeypot. Throughout the whole ever changing era in worldwide data exchange, fast data generation as well as cheap access of internet, this was as highest concern for emphasize home network protection throughout today's world otherwise else we might be exposed to hackers who aim to manipulate our digital environment for evil motives. An IDS, Honeypot and packet analyser as network management tool may be an incredibly beneficial addition to one's small or home network for security.

II. BACKGROUND

A. Raspberry Pi

Using the shortest possible terminology, Raspberry Pi is a single board device invented by Eben Upton. One of the founders of Raspberry Pi, Eben Upton created the first Raspberry Pi in an attempt to solve the problem of a declining student population in computing science at the university. It was created to provide an inexpensive device to the public that could increase programming and hardware skills among young people and along to encourage the students to fundamental education of computer technology in schools and universities. The computing capacity of the Raspberry Pi is obviously much less than a standard machine but is still a compact, low-cost, Linux-based device capable of making all the programs running at a reduced power consumption level. Figure 1 displays concept of Raspberry Pi 4 architecture. A cost-effective, compact mini-computer that can be inserted into an output display, along with a regular keyboard and mouse to completely act as a typical desktop or laptop.

It is an versatile compact tool that is capable of performing anything you would want to do so on a standard computer, from surfing around the internet, playing high end video games, designing cost effective and power effective home automation systems. This system often has the potential to communicate with the outer world and is used widely in nearly every area for a number of projects.

Fig. 1. Raspberry Pi 4 Model B Board

B. Intrusion Detection System (IDS)

An Intrusion Detection System is a software application or hardware application that will store and evaluate the data by intercepting inbound and outbound network traffic as per specific rules it will detect the suspicious activities and alerts when such activity is discovered. Intrusion detection systems are an important part of network security because they carry out comprehensive network protection strategies by detecting and preventing threats accompanied by data storing for potential review.

An IDS can be classified based on types:

1) **Network Based IDS:** A NIDS whereas installed on a physical network and where the inbound and outbound traffic to and from all network devices can be monitored.
2) **Host Based IDS:** A HIDS whereas installed on a registered network administrators, or computers. A HIDS only checks incoming and outgoing system packets and warns the owner whenever unusual and harmful behavior were observed. This collects information from current device archives and links them to previously collected data.

An IDS can be classified based on detection approach:

1) **Signature Based detection:** Signature based or rule based detection that monitors all the data passing throughout their infrastructure and distinguish them against the collection of database that contain incident signatures for detecting the threats. This method utilizes standards or files published from or held by security experts.
2) **Anomaly Based detection:** Detection based on deviation or activity tracks the traffic of networks, servers and consumers across time to collect evidence and afterwards produce warnings whenever unusual or irregular activity has been observed. All of this utilizes rules that are based on attacks observed.

C. Honeypot

A honeypot would be implemented on network which act as decoy designed for attracting attackers and collect details about both the intruder and the attack. A honeypot emulates networks in simplistic terms that could have the ability to be targeted by hackers. A honeypot's key purpose is to avoid attacks and collect data for more research work about the threat.

The mechanism of security of a honeypot which produces the simulated maze for attracting attackers. The deliberately infected operating program helps an intruder to discover bugs so that researcher can research them and strengthen the protection policies. A honeypot may be applied on file servers and routers from applications and networks to any computing resource. They are able to maintain specifics record of the attack using automated resources to record and track tools. The honeypot issued the ip address, which is continuously monitored by administrator. Honeypot must wait before an interaction with the machine is initiated, and any kind of contact with the honeypot is considered as suspicious. They might very well waste the time of the intruder, if not absolutely redirect the intrusion and, if correctly implemented, can be highly successful.

When deployed in a network, a honeypot will collect as much evidence as possible regarding the attack, which will help in the analysis and removal of system weaknesses when analyzed, thus eliminating any security loop holes and keeping the system secure.

They may also be combined with software to generate alerts and to log output. They can also be combined with software to generate alert and to log output. Based on its deployment, a honeypot may be a research honeypot or a productive honeypot.

There are two primary types of honeypot based on deployment method:

1) **Research Honeypot:** Research honeypot mainly used by non-profit organizations and educational institutions for the specific purpose of researching the hacker community's motivations and strategies for targeting various networks.
2) **Production Honeypot:** Production honeypot mainly used by companies and corporations to investigate the motivations of hackers, as well as to minimize and reduce the possibility of threats on the network as a whole.

Honeypots can be further classified as:

1) **High Interaction Honeypot:** Those are the most sophisticated honeypots although they are complicated and hard to implement. Honeypots of this kind have their own operating system. It is indeed a mixture of decoys all of which function as someone with various degrees of interaction. They are therefore higher-maintenance and need skills through the use of advanced technology such as virtual machines to guarantee hackers will be unable to reach the actual network.
2) **Medium-Interaction Honeypot:** This honeypots that can trick the intruder appears to be running a bogus operating system and therefore lures the intruder.
3) **Low Interaction Honeypot:** This form of honeypot is the most representative widely found in a production community.

   Low-interaction honeypots run a handful of services and function more than anything as a tool for early warning detection. Like several security teams deploying numerous honeypots across various aspects of their network, they are easy to deploy and manage.

D. Packet Analyzer

The network packet analyzer operates by acquiring and further analyzing and decoding real-time network traffic to provide useful data that can be used in identify implicit network issues. Their track and display the inbound and outbound traffic of the network in a human usable form. They track and view the inbound and outbound traffic on the network in a human usable form. A packet capture is a specialized hardware system installed on a network, or an application security program operating on a dedicated system.

Monitoring systems can support many network control devices, such as anti-malware and firewalls, which play a vital role in network management and security. Analysts can, however, be deployed both illegally and legal. An analyzer designed to provide network protection, installed with the owner's permission to provide troubleshooting and to control a network can be considered a legitimate sniffer, but analyzers used by hackers to obtain unauthorized access with the purpose of monitoring or stealing confidential data without the owner's permission are considered an unlawful sniffer. Packet analyzers which mostly are not identified because they passively gather network data, although they are likely to be identified by the active analyzers.
Packet analyzers capture and monitor network information that is available on the network interface to which they are installed. Based on the network analyser configured, the data can be retrieved depends on it. The data collected by the analyzer in a wired network is based on the architecture of the network, depending on the network switch setup. Analyzers will only collect data from one channel at a time in a wireless network, unless multichannel capturing is allowed on multiple network interfaces. When the data is collected it will be delivered for analysis in a human usable form. Data is used to assess defects and weaknesses, such as detecting unanswered requests from the network.

III. PROPOSED SYSTEM

The proposed system consists of basic setup of hardware and the software. The hardware we are using is Raspberry Pi 4B module which we will connect to Wi-Fi router with wireless connection or connect with LAN cable. First we will setup the hardware environment by preparing setup. We require monitor for display, raspberry pi module with HDMI cable for displaying output, memory card booted with operating system and external mouse and keyboard. So now we have connected raspberry pi module using HDMI interface for output along with mouse and keyboard to the USB ports for input. It will be operated by the use of a micro-usb cable. The raspberry pi memory card has been boot-loaded with a Raspbian-Stretch-2018-Desktop-Image. And normally install the operating system in raspberry pi module, connect to internet and then before installing other packages update and upgrade the Raspbian OS packages.

As shown in Figure 1 there would be an intrusion prevention program, a packet analyzer and a honeypot application across the router access point and the local network. We also built an intrusion detection system based on the network, a packet sniffer-analyzer and a decoy honeypot. The server is linked to several clients owing to the centralization of the received data and is configured to accept all incoming messages and are stored in the information database.

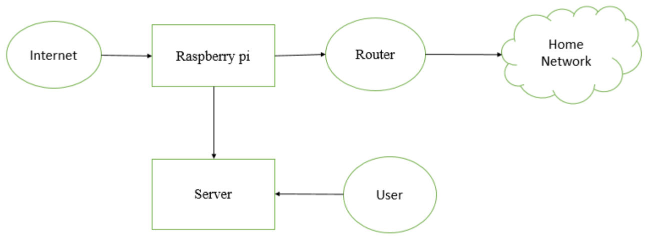


Fig. 1 Architecture of a system

The DHCP setup for the eth0 interface was provided to the Raspberry Pi, and the eth0 interface was allowed at boot if we are using LAN cable as Ethernet.

A. *Intrusion Detection System*

On the Raspberry Pi we are configuring a Network Intrusion Detection Program focused as guidelines. To customize our IDS we use the following Open Source software:

1) Snort: The open source network intrusion detection applications which is snort is one most commonly used and widely deployed designed to provide real-time monitoring and sniffing of network traffic on communication networks. This was originally designed in 1998 by Martin Roesch, currently managed by Cisco Systems. As a packet-analyzer snort is prepared to act, pre-processor, and engine-logger detection but most of all, snort is safe and functional with many architectures. The collected data packets in real-time which are stored for future study in the' tcpdump' layout. Snort is known to provide a large range of identities for malicious attacks. It is searching for and recording a specified collection of data in the packet stream, based on the collection of defined rules and signatures. The packets sniffed from the network are decoded by the packet decoder and sent to the pre-processor where the job seems to be to change it to satisfy the specifications of the detection engine. Based on the collection of rules and signatures installed, it searches for a specified range of data in the packet stream and records them. The packet-Decoder will decode the sniffed packets from the network and is then forwarded to the pre-processor whose job seems to be modify them to fulfil the detection engine requirements. The monitoring engine scans the packets for any suspicious behavior to occur. The monitoring device records unauthorized packets which can be harmful.

2) Barnyard2: The snort output module done by barnyard2 which will translates snort-generated warnings into the format of a database. It is an open source snort translator and assists in data analysis. Unified2 is a log file where snort stores the sniffed packets into a binary format as a log file. Barnyard2 has the task of gathering this data and converting in a readable format accompanied by submitting it to specific database as logging repositories. The two main categories of specifications are input processors and output plugins. The reading of data from unified2 file format is a work of Input processors, then output plugins will record the output.

3) PulledPork: Pulledpork is a PERL-based rule creation method for intrusion detection programs, such as Snort. This can be used to identify you edition of Snort, and to create an acceptable rule set for you automatically. The term ' PulledPork' was clearly selected since this program is pulling the new rules.

Figure 2 demonstrates the interplay of Snort, Pulledpork and Barnyard3.

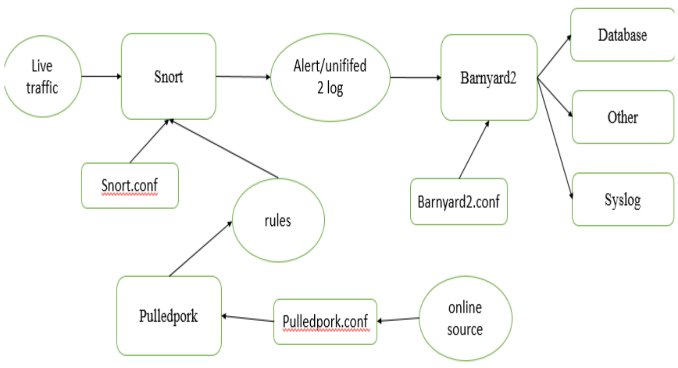


Fig. 2 Architecture of IDS

B. Honeypot

With our Raspberry Pi honeypot, we are configuring a medium-level SSH honeypot interface, Cowrie, used to monitor full shell interface during threat along with SSH and brute force attempts.

1) Cowrie: This is medium level interface honeypot developed as a python-based, that might quite easily monitor this whole shell activity throughout an attempt, including the ability to record brute force actions, referred to as brute force cracking. It is a tedious frustrating trial and error-based technique utilized by criminals for bypass passwords and paraphrases. This honeypot helps any intruder to start a communication then sign further into the network, conning themselves into assuming they are accessing their legal SSH session of the device. When the brute-force attempt to break the authentication has been completed, the intruder would be routed to the bait interface they will communicate through. If the intruder reaches this bait network, the network imitates the real device and is able to track and record any harmful behavior that the intruder is undertaking. Cowrie provides artificial file systems where data can be inserted then removed through successful imitation. This still preserves most of the data accessed during the connection.

2) Dionaea: It’s a Low level Interaction python based honeypot. Dionaea's purpose is to capture malware leveraging weaknesses revealed by networked services. The honeypot's primary aim is to obtain a duplicate of the sample malware. Honeypot dionaea has implemented on environment to detect and collect malware attacks by disclosing the malwares to external monitoring service providers so that malware can be identified until it is become a critical threat in the modern information technology world. It will store all the capture malware sample which is detected by the raspberry pi module and create the logs store in honeypot as a databases.

C. Packet Analyzer

We're going to install Tshark, the Wireshark as command line version for packet capture as well as evaluation, to set up our Raspberry Pi as the network sniffer as well as analyser.

1) Tshark: Tshark is a terminal-based network management and reporting application utilized for tracking the traffic on the network as real time. This is a network analysis device used to capture and evaluate the traffic flow through networks of connection. This helps anyone could gather network information in real time even review packets through subsequently collected network logs and view them as a user friendly environment. The software which Tshark uses to collect data as pcap and tcpdump. If neither choices are being provided to Tshark then it should operate such as tcpdump.

Packet sniffers could operate like an important network protection and surveillance device. As starters, these were used to track infrastructure activity, for determine when a network becomes overloaded and thus to detect inefficiencies. In fact, these might be used to classify network vulnerabilities and to find vulnerability weaknesses, as example, when a network fault happens, they are being used to determine the cause of the mistake. Tshark catches packets that, in effect, expose packet details, layer details, free memory information and offer a schematic description for everything those details mentioned.

IV. IMPLEMENTATION

We implement the honeypot on top of a hardware module (Raspberry Pi) which is less expensive than the modern system like some independent separate external device in a small infrastructure. All the logs or the captured data either stored locally as well as in cloud server for future analysis.

As we know honeypot would be implemented like an fake network designed for attracting the malicious users then gathering information regarding both intruder and target and also defend our network. So we deploy the Intrusion Detection System software on a raspberry pi module from the official source. Snort is the best open source tool for capturing and alerting the user as well as the administrator. After deploying the snort we will modify the snort.config for the changes to occur in snort. We can either create our own rule set or we can get rules set from the sources as per users choice. After deploying snort we can deploy barnyard2 as a snort output component then process the snort created warnings towards a database layout which we can view in human readable format and the format can be customized. While the barynard2 running it will create the log file as unfified2 as a output by snort stored log files. After that we will deploy Pulledpork is a PERL-based rule creation method for intrusion detection applications such as Snort and Suricata. This will be configured for identify your variant of Snort then instantly create an appropriate collection of rules required it. If the user couldn’t able to configure rule set manually then pulledpork can be useful which is available in snort sources. By simply deploying the pulledpork and running the command can able to check the version of rule set and if older database it will update automatically. All the tool like snort, barynard2 and pulledpork are act as a complete IDS and each time we have to start manually, so instead starting manually we can write small startup shell script which will running automatically as soon as raspberry pi bootup.

We can check whether IDS is working by checking the services running in the module and all the honeypot logs will be stored in /var/log/IDS_name.

Secondly, we will implement medium level interaction honeypot on a raspbain operating sytem. Honeypot cowrie, that might quite appropriately monitor that whole shell activity throughout most of the operation, including the ability to document brute force actions, known to as brute force cracking. For deploying the cowrie we will download and extract the file and check the dependency required to install before installing. After that we will install the cowrie and run the honeypot as a startup by creating the shell script and check the status. As it running if honeypot discovered any suspicious traffic it will alert the user by sending the mail or notification. And other honeypot is dionaea which will be installed in raspberry pi module for detecting and capturing the malware in the network. It will store the capture malware logs and create database of all captured malware and can be shared globally in the cloud server for analysis. Logs will be stored in /var/log/dionaea

Thirdly, we will implement tshark which is the terminal-based network tracking and evaluation framework used for real-time data traffic monitoring. We will download the tshark file from official source and install in the raspberry pi module and run the tool which will monitor all the network traffic in real time and create the logs and alert the user.

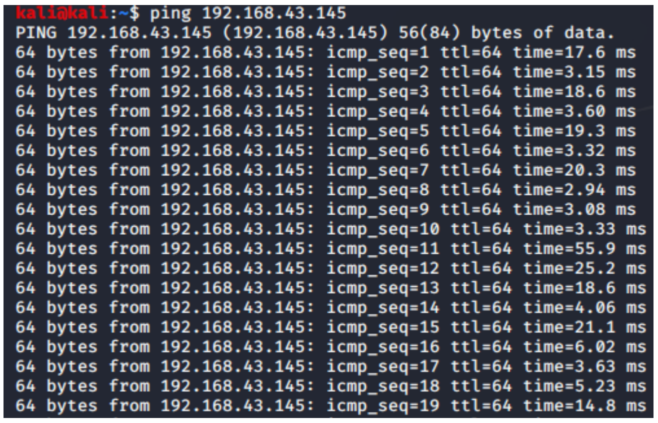
Finally, we can store logs in two ways either locally or in the server. Logs which will be created by every IDS and honeypot will be stored in /var/logs/ folder. For the server we have to create server which is implemented in Ubuntu OS on top of System. Here, we will be using MHN server installed on top of Ubuntu OS that will stored all the logs captured b the raspberry pi and show in the human visual format. For deploying MHN server, server need to downloaded and installed on Ubuntu OS and can be configured either in cloud or Virtual Machine. Here, we have installed in virtual machine and open the server through the browser.

V. RESULT

To effectively check ours suggested IDS module, Packet Analyzer module and honeypot program, the successful running of modules was assured.

A. Output of Intrusion Detection System

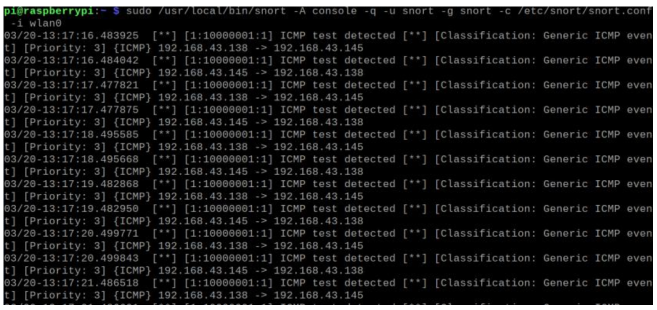
The Raspberry Pi has been checked by the offensive device for testing whether warnings were recorded and captured by snort. Figure 3 displays the ping instruction the attacker computer is performing.



```
kali@kali:~$ ping 192.168.43.145
PING 192.168.43.145 (192.168.43.145) 56(84) bytes of data.
64 bytes from 192.168.43.145: icmp_seq=1 ttl=64 time=17.6 ms
64 bytes from 192.168.43.145: icmp_seq=2 ttl=64 time=3.15 ms
64 bytes from 192.168.43.145: icmp_seq=3 ttl=64 time=18.6 ms
64 bytes from 192.168.43.145: icmp_seq=4 ttl=64 time=3.60 ms
64 bytes from 192.168.43.145: icmp_seq=5 ttl=64 time=19.3 ms
64 bytes from 192.168.43.145: icmp_seq=6 ttl=64 time=3.32 ms
64 bytes from 192.168.43.145: icmp_seq=7 ttl=64 time=20.3 ms
64 bytes from 192.168.43.145: icmp_seq=8 ttl=64 time=2.94 ms
64 bytes from 192.168.43.145: icmp_seq=9 ttl=64 time=3.08 ms
64 bytes from 192.168.43.145: icmp_seq=10 ttl=64 time=3.33 ms
64 bytes from 192.168.43.145: icmp_seq=11 ttl=64 time=55.9 ms
64 bytes from 192.168.43.145: icmp_seq=12 ttl=64 time=25.2 ms
64 bytes from 192.168.43.145: icmp_seq=13 ttl=64 time=18.6 ms
64 bytes from 192.168.43.145: icmp_seq=14 ttl=64 time=4.06 ms
64 bytes from 192.168.43.145: icmp_seq=15 ttl=64 time=21.1 ms
64 bytes from 192.168.43.145: icmp_seq=16 ttl=64 time=6.02 ms
64 bytes from 192.168.43.145: icmp_seq=17 ttl=64 time=3.63 ms
64 bytes from 192.168.43.145: icmp_seq=18 ttl=64 time=5.23 ms
64 bytes from 192.168.43.145: icmp_seq=19 ttl=64 time=14.8 ms
```

Fig. 3 Ping performed by the attacker system

The ICMP specifics have been entered into the Raspberry Pi terminal, as seen in figure 4. ICMP ping requests created by the attacker machine were successfully logged by Snort Intrusion Detection System.

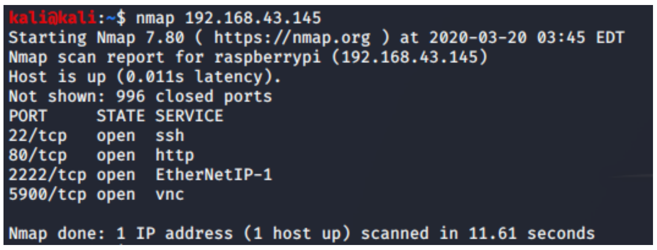


```
pi@raspberrypi:~ $ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf
 -i wlan0
03/20-13:17:16.483925  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
03/20-13:17:16.484042  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.145 -> 192.168.43.138
03/20-13:17:17.477821  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
03/20-13:17:17.477875  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.145 -> 192.168.43.138
03/20-13:17:18.495585  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
03/20-13:17:18.495668  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.145 -> 192.168.43.138
03/20-13:17:19.482868  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
03/20-13:17:19.482950  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.145 -> 192.168.43.138
03/20-13:17:20.499771  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
03/20-13:17:20.499843  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.145 -> 192.168.43.138
03/20-13:17:21.486518  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP even
t] [Priority: 3] {ICMP} 192.168.43.138 -> 192.168.43.145
```

Fig. 4 Display the capturing of packets by IDS

B. Output of Honeypot Analysis

We'll run a nmap scan to check Cowrie, accompanied by a brute force assault from the Kali-Linux systems. Nmap Scan: Scanning for open-port weaknesses in the target device is done, as seen in figure 5.



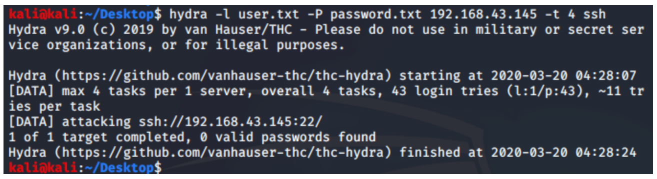
```
kali@kali:~$ nmap 192.168.43.145
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-20 03:45 EDT
Nmap scan report for raspberrypi (192.168.43.145)
Host is up (0.011s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
2222/tcp open  EtherNetIP-1
5900/tcp open  vnc

Nmap done: 1 IP address (1 host up) scanned in 11.61 seconds
```

Fig. 5 Display the weakness capture by scanner

Now we will try to use SSH terminal to reach any available ports, accompanied through the brute-password.

The Brute Force Attack employs the trial and error approach for collect sensitive client knowledge, along with username and passwords that are often used for unauthorized entry, like those seen in Figure 6.

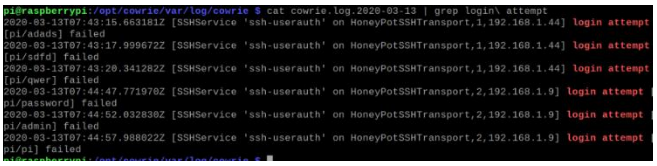


```
kali@kali:~/Desktop$ hydra -l user.txt -P password.txt 192.168.43.145 -t 4 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret ser
vice organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-03-20 04:28:07
[DATA] max 4 tasks per 1 server, overall 4 tasks, 43 login tries (l:1/p:43), ~11 tr
ies per task
[DATA] attacking ssh://192.168.43.145:22/
1 of 1 target completed, 0 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-03-20 04:28:24
kali@kali:~/Desktop$
```

Fig. 6 Shows the brute force attack attempt

During the attack, the cowrie logs were reviewed then, like those seen in Figure 7, Cowrie were effective at identifying and recording the SSH Brute Force Attack.



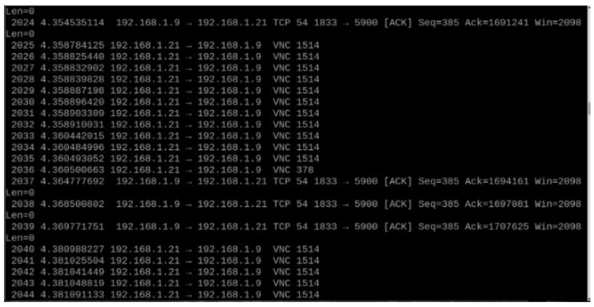
```
pi@raspberrypi:/opt/cowrie/var/log/cowrie $ cat cowrie.log.2020-03-13 | grep login\ attempt
2020-03-13T07:43:15.663181Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,1,192.168.1.44] login attempt
[pi/adads] failed
2020-03-13T07:43:17.999672Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,1,192.168.1.44] login attempt
[pi/sdfd] failed
2020-03-13T07:43:20.341282Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,1,192.168.1.44] login attempt
[pi/qwer] failed
2020-03-13T07:44:47.771970Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,2,192.168.1.9] login attempt
pi/password] failed
2020-03-13T07:44:52.032830Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,2,192.168.1.9] login attempt
pi/admin] failed
2020-03-13T07:44:57.988022Z [SSHService 'ssh-userauth' on HoneyPotSSHTransport,2,192.168.1.9] login attempt
pi/pi] failed
```

Fig. 7 Capturing of attack attempted logs on raspberry pi

C. Output of Packet Analyzer

To catch network traffic of all sorts, the Tshark Packet Analyzer was launched with the corresponding instructions.

Figure 8 demonstrates all the Tshark effectively collected network traffic data packets



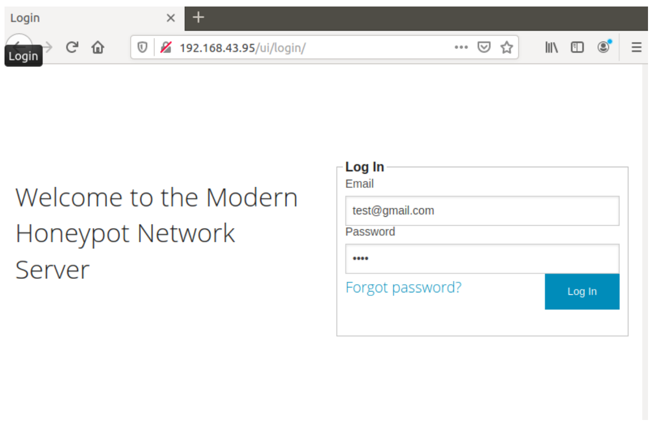
```
Len=0
 2024 4.354535114  192.168.1.9 → 192.168.1.21 TCP 54 1833 → 5900 [ACK] Seq=385 Ack=1691241 Win=2098
Len=0
 2025 4.358784125 192.168.1.21 → 192.168.1.9  VNC 1514
 2026 4.358825440 192.168.1.21 → 192.168.1.9  VNC 1514
 2027 4.358832902 192.168.1.21 → 192.168.1.9  VNC 1514
 2028 4.358839828 192.168.1.21 → 192.168.1.9  VNC 1514
 2029 4.358887198 192.168.1.21 → 192.168.1.9  VNC 1514
 2030 4.358896420 192.168.1.21 → 192.168.1.9  VNC 1514
 2031 4.358903309 192.168.1.21 → 192.168.1.9  VNC 1514
 2032 4.358910031 192.168.1.21 → 192.168.1.9  VNC 1514
 2033 4.360442015 192.168.1.21 → 192.168.1.9  VNC 1514
 2034 4.360484996 192.168.1.21 → 192.168.1.9  VNC 1514
 2035 4.360493052 192.168.1.21 → 192.168.1.9  VNC 1514
 2036 4.360500663 192.168.1.21 → 192.168.1.9  VNC 378
 2037 4.364777692  192.168.1.9 → 192.168.1.21 TCP 54 1833 → 5900 [ACK] Seq=385 Ack=1694161 Win=2098
Len=0
 2038 4.368500802  192.168.1.9 → 192.168.1.21 TCP 54 1833 → 5900 [ACK] Seq=385 Ack=1697081 Win=2098
Len=0
 2039 4.369771751  192.168.1.9 → 192.168.1.21 TCP 54 1833 → 5900 [ACK] Seq=385 Ack=1707625 Win=2098
Len=0
 2040 4.380988227 192.168.1.21 → 192.168.1.9  VNC 1514
 2041 4.381025504 192.168.1.21 → 192.168.1.9  VNC 1514
 2042 4.381041449 192.168.1.21 → 192.168.1.9  VNC 1514
 2043 4.381048819 192.168.1.21 → 192.168.1.9  VNC 1514
 2044 4.381091133 192.168.1.21 → 192.168.1.9  VNC 1514
```

Fig. 8 Capturing of network packets in network

D. Output of MHN Server

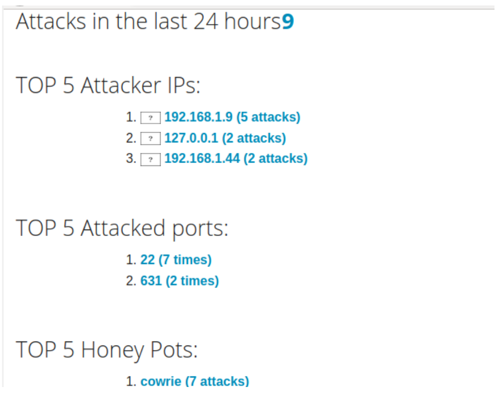
The logs of IDS, honeypot are all will be sent to the MHN Server for Hyman visual format and create the graphs for analysis which make easy to understand for non-technical users.

Figure 9 and 10 demonstrates all the log capture by the honeypot, IDS and tshark were displayed in human visual format for better understanding attack happened in the network.



Welcome to the Modern Honeypot Network Server

Fig. 9 Data logging on the server



Attacks in the last 24 hours9

TOP 5 Attacker IPs:

1. 192.168.1.9 (5 attacks)
2. 127.0.0.1 (2 attacks)
3. 192.168.1.44 (2 attacks)

TOP 5 Attacked ports:

1. 22 (7 times)
2. 631 (2 times)

TOP 5 Honey Pots:

1. cowrie (7 attacks)

Fig. 10 Shows the overall information

VI. Conclusion

Security is unquestionably one of our generation's greatest concerns. When deployed properly, devices including the intrusion detection devices, honeypots and packet analyzers may be used quite effectively can tackle against security threats may be used quite effectively to tackle against security threats and maintain data privacy and protection. Using even raspberry pi as a network monitoring system for protecting against small network is less expensive then deploying the monitoring system in a single system.

In this paper we have implemented the devices which will monitor the small network by proposing raspberry pi 4 deployed with IDS and honeypot as a detection alert system. Whereas snort as IDS, cowrie and dioneae as a honeypot and packet analyzer as a Tshark. Raspberry pi 4 board installation and implementation as a honeypot led to low budget and also guarantees functionality as well. In future instead of configuring the server in local, we can deploy in cloud and can be accessed or monitor remotely.

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. M. Cosar and S. Karasartova, “A firewall application on SOHO networks with Raspberry Pi and Snort,” 2017 International Conference on Computer Science and Engineering (UBMK), 2017.
3. D. Robinson and C. Kim, “A cyber-defensive industrial control system with redundancy and intrusion detection,” 2017 North American Power Symposium (NAPS), 2017.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. A. Ranjan, S. Sinha, and S. Swain, “Signature Based Wireless Intrusion Detection Using Raspberry Pi,” 2016 International Journal of Computer Engineering and Applications, Volume X, Special Issue, ICRTCST, 2016.
10. C. Bousaba, T. Kazar, and W. Pizio, “Wireless Network Security Using Raspberry Pi,” 2016 ASEE Annual Conference & Exposition Proceedings, 2016.

AUTHORS PROFILE

Ms Monica Catherine, working as assistant professor in the department of Information Technology, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu

Mr Ashok Kumawat persuing master of technology in Information security and cyber forensic, SRM Institute of Science and Technology, Kattankulathur**,** Tamil Nadu.