Hybrid H-DOC: A bait for analyzing cyber attacker behavior

Original Scientific Paper

Amal M. R.
Noorul Islam Centre for Higher Education,
Research Scholar, Department of Computer Science and Engineering
Kumarakoil, Tamil Nadu, 629175, India
amalmr589@gmail.com

Venkadesh P
Noorul Islam Centre for Higher Education,
Assistant Professor, Department of Computer Science and Engineering
Kumarakoil, Tamil Nadu, 629175, India

Abstract *– Cyber security is a vital concern for companies with internet-based cloud networks. These networks are constantly vulnerable to attack, whether from inside or outside organization. Due to the ever-changing nature of the cyber world, security solutions must be updated regularly in order to keep infrastructure secure. With the use of attack detection approaches, security systems such as antivirus, firewalls, or intrusion detection systems have become more effective. However, conventional systems are unable to detect zero-day attacks or behavioral changes. These drawbacks can be overcome by setting up a honeypot. In this paper, a hybrid Honeynet model deployed in Docker (H-DOC) bait has been proposed that comprises both low interaction and high interaction honeypot to attract the malicious attacker and to analyze the behavioral patterns. This is a form of bait, designed to detect or block attacks, or to divert an attacker's attention away from the legitimate services. It focuses only on the SSH protocol, as it is widely used for remote system access and is a popular target of attacks. The proposed Hybrid H-DOC method identify ransomware activity, attack trends, and timely decision-making through the use of an effective rule and tunes the firewall. The attack detection accuracy of the proposed Hybrid H-DOC method when compared with IDH, Decepti-SCADA, AS-IDS and HDCM is 13.97%, 11.82%, 8.60% and 5.07% respectively.*

Keywords*: cybersecurity, docker, containers, high interactive honeypots, low interactive honeypots*

1. INTRODUCTION

Cyber security entails the protection of resources that are connected to the Internet. Every day, malicious activities on the Internet is becoming more common [1]. Cyberattacks are on the rise at an exponential rate, as millions of attacks are identified annually, requiring more complex and automated analysis techniques because existing technology cannot handle the volume of data or the diversity of attacks [2]. Cyber risks are complicated and time-consuming to understand and address. The analysis of honeypot data can identify cyber threats [3].

Honeypot technology is a dynamic and ever-growing technology [4]. Honeypot is a network computer and server configured such that to appear vulnerable and to interact highly to attackers, to attract the attackers with open flaws and known vulnerabilities [5]. In computing security, honeypots are frequently used by scientists and security specialists, depending on their level of involvement. There is a unique feature of honeypots that makes any communication with them illegitimate because they are not providing any genuine services [6]. Real world scenario of Honeypot is shown in Fig. 1.

Generally, honeypots are used and are widely distributed. However, there are a number of difficulties that need to be addressed. Security, flexibility, and a limited number of IP addresses are all factors to consider [7, 8]. Honeypots are also prone to potential attackers avoiding them because of their nature. Therefore, it was decided to use operating system level virtualization, otherwise known as containerization, to execute the selected honeypots [9,10]. Containers are virtual environments in which a program and its dependencies are packaged together [11]. The operating system and kernel can be shared by containers, making them less resource-intensive than virtual machines [12-14].

In addition, by running in user space, they minimize system burden [15,16].

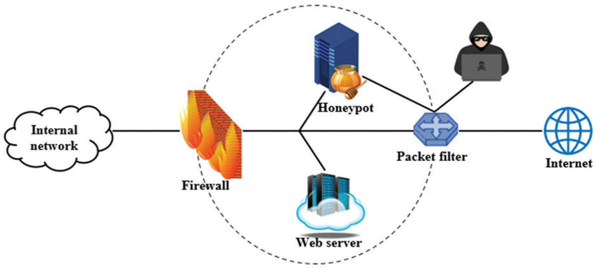

Fig. 1. Real world scenario of Honeypot

A Docker container is used in this experiment. In addition to x86-64 and ARM, Docker supports numerous architectures natively, meeting the compatibility criterion. By using Docker Compose, the load balance on the target system were also managed and create, deploy, and orchestrate the instances using the API easily and quickly [17,18]. This study suggests a hybrid honeynet model implemented in Docker. Additionally, the security of Honeypot implementations within Docker containers is investigated.

The rest of the paper is organized in the following manner. Section II represents the literature review in detail. Section III describes the Hybrid H-Doc bait in detail. Section IV describes the security analysis. Section V describes the conclusion and future work.

2. LITERATURE REVIEW

The history of cyber security experimentation (CSE) platform was traced back to the early 21st century. Due to the cumulative amount of cyberattacks, countries are investigating the development of a CSE platform. (IDS) Intrusion Detection System, (VDS) Vector Deep Surveillance and honeypot system. Among the above said CSE platforms honeypot is the highly efficient scenario. The "state of the art" of present honeypot solutions is presented in this section.

In 2018, Almohannadi, H., et al. [19] proposed a new threat intelligence technique that evaluates honeypot log data to identify attacker behaviour and find attack trends. They've set up a honeypot on an AWS cloud to collect cyber incident log data in order to achieve this goal. Elasticsearch technology, specifically an ELK (Elasticsearch, Logstash, and Kibana) stack, is used to analyze the log data.

In 2019, Yin, et al.,[20] present a new architecture for a cyber security experimentation platform on the basis of Docker. This software has the scalability and flexibility necessary for large-scale cyber simulations. This feature enables users to customize cybernode's topologies, software environment, and also support the customization of important experiment indicators. It is possible to transmit important experiment indications in real-time, thereby reducing the total cost and facilitating the analysis process.

In 2021, Buzzio-Garcia, J. [21] suggests the utilization of Docker as a high-interaction honeypot, so that threats can be detected at both the network and host levels. It was developed using open-source tools to ensure scalability, safety, and dynamic functionality. A real-world test has demonstrated that it is capable of capturing harmful data for examination at the network and host level employing tools like VirusTotal.

In 2022, Sivamohan, S., et al [22] used Docker container technology with a honeynet-based IDS to create an efficient active protection architecture. The creation of honeynet technology is crucial to cloud security and threat detection. Based on the results of the experiment, it appears that this defense system can identify and log the attacker's activities, revealing new attack strategies and even zero-day vulnerabilities.

From the existing methods, it is identified that, there is no solution focused solely on the SSH protocol. An SSH connection encrypts connections between two end points and provides password or public-key authentication. A secure alternative to unsecure file transmission methods and legacy login protocols (such as telnet and rlogin) (such as FTP). The position of authorized key files and port forwarding in SSH, however, are not ideal. Port forwarding allows an attacker to get around firewalls that have been set up to restrict access to the server's network. Due to their encrypted SSH connection, the attackers are undetectable. So, it is important to focus on the SSH protocol in order to reduce the above-mentioned issues. Therefore, in this paper a hybrid H-Doc bait has been proposed which concentrates on the SSH protocol.

3. PROPOSED HYBRID H-DOC BAIT

In this research, an attacker's behavior as well as metadata are utilized to address the problem. The procedure involved in this research process for implementation are as follows:

- Setting Up EC2 instance
- Implementing Docker on Cloud
- Setting Up the Hybrid Honeynet in Docker
- Tuning the firewall

3.1. SETTING UP EC2 INSTANCE

Among the most popular instances are General Purpose ones, which are an excellent way to get started with AWS or cloud computing. Their most common uses include web servers, development environments for mobile apps, and enterprise applications such as CRMs and ERPs. Within this class, the most important distinction is between instances that have a Fixed performance and those that have a Burstable performance. Using burstable performance EC2s, one can easily grow their computational power.

3.2. IMPLEMENTING DOCKER ON CLOUD

A Docker container has several advantages, including its ability to be deployed in development, test, staging, and production environments, and its ability to be integrated into distributed systems. Applications are constructed with Docker technology, which is delivered to end users via AWS EC2 services. Scalability and management of Docker containers are best handled by Amazon EC2 Container Service. Docker containers with the EC2 Container Service are used to run processes on Amazon EC2 instances using optimistic, shared state scheduling. Amazon ECS allows you to run containers across many hosts, isolate applications and users, and scale quickly to meet your applications' and users' changing needs.

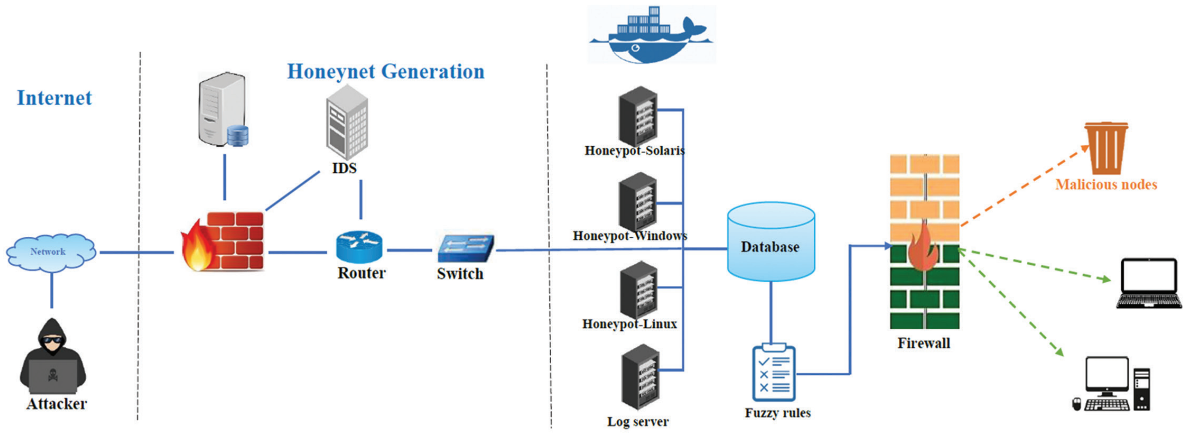

Fig. 2. Proposed framework

3.3. SETTING UP THE HONEYPOT IN DOCKER

Package the honeypot environment in a docker container to deploy it in various nodes which contain docker for faster deployment mainly in UNIX based distros.

3.4. TUNING THE FIREWALL

Real-time data flows will be handled by a uniform, high-throughput, low-latency platform developed by the project. There are tools for processing continuous and timely events as well as extracting high-level knowledge events from lower-level events, is known as complex events. An inference engine uses working memory and fuzzy rules to make decisions. Apache Kafka has been used, which is a big data processing tool. This can process big streams of data. There is a fuzzy rule base that comprises all the rules, and working memory keeps track of the most recent state of the system. As soon as a feature extraction packet is received, it is sent to an inference engine for identification; if they are considered attacks, the firewall blocks them.

Fig. 2 represents the overall framework of the proposed method. When the attacker tries to enter the network through SSH, it will be assigned to hybrid honeypots which contains both low interaction and high interaction honeypots. Unused and unwanted containers will be removed through container removal. Loggings contains the visiting informations of the attacker.

3.5. ATTACK ENTRY VIA SSH

The purpose of this study is to investigate into SSH connections to honeypots from any IP address, usually over port 22. By using the honeypots, the attacker is given access to a Linux shell console. Devices with IP addresses that connect to a honeypot are considered attackers. This paper defines a session as any SSH connection between an attacker and a honeypot that is approved by the honeypot. A honeypot is attacked by connecting to an SSH port, usually port 22, and establishing an SSH protocol session with the attacker. During the session, the attacker can enter commands, download and execute files, and so on, to communicate with the honeypot.

3.6. SSH SCENARIO

An SSH session was established with a sophisticated simulated attacker. The following was found:

- The traffic was first forwarded to the low interaction honeypot after installation.
- Expert system assesses whereas, to send the traffic to high interaction honeypot or to stay in the low interaction honeypot system. After connection, the attacker can navigate to the honeypot terminal with the fake file system.
- By using the command 'vi,' a fingerprinting attack quickly identified a popular fingerprint indication for honeypots.
- The honeypot container logs recorded all interactive contacts with the attacker session, which were stored and sent to syslog.

3.7. HYBRID HONEYNET MODEL

- Honeypots are grouped into clusters called Honeynets to prevent them from becoming independent units which is shown in Fig. 3. The benefits of such setups include Real-time correlation of sensor data, One point of sensor control, Central storage of event data. However, such a distributed model has a few drawbacks Complexity of infrastructure, greater security, risk Inefficient management that should be considered as well. In order to overcome the disadvantages, hybrid honeynet has been proposed.

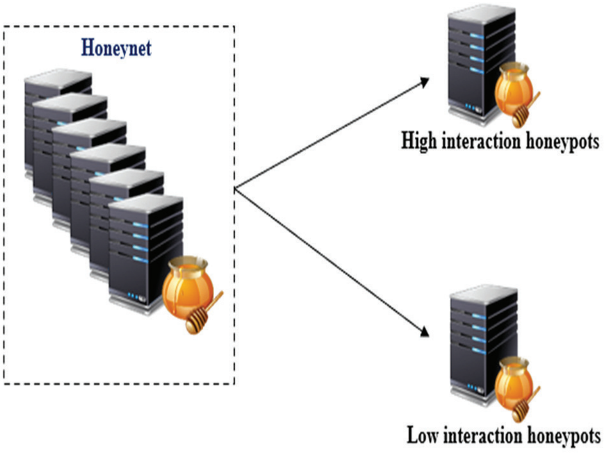

Fig. 3. Schema of hybrid honeynet

3.8. LOW INTERACTION HONEYPOT (LIH)

During aggressive expansion, low-interaction honeypots are simple, yet can save time due to intruder detection, and the honeypot imitate can be reduced with specific commands. Honeyd, meanwhile, is a honeypot with a low-interaction level. In order to imitate services with low interaction, attackers can take advantage of the low interaction honeypot. Because of the minimal amount of contact, this type of honeypot gathers data from the first step of an attack. Information about the threat's reason for attacking is seldom obtained.

3.9. HIGH INTERACTION HONEYPOTS (HIH)

Honeypots with high interaction are the opposite of honeypots with low interaction in deception technology. In contrast to merely simulating particular protocols or services, the attacker actually attacks real systems. This makes it less likely that they will understand they are being monitored or diverted. Since these systems are only available as decoys, all communications discovered are hostile by its very nature, simplifying it to finding threats and track an attacker's activity. "Lyrebird" is a honeypot framework that is highly interactive. All communications between the attacker and the computer are recorded in clear text, so the attacker has access to real vulnerable programs. The Schema of the honeynet with a single Low and high interaction honeypot is shown in Fig. 4.

3.10. REQUIREMENTS OF THE SERVER WITH EXPERT SYSTEM

Sessions should satisfy one of two hypotheses:

- The sessions could be diverted to a LIH
- The sessions could be diverted to a HIH
- To make decisions, you must use the data provided by the simulated environment, low interaction honeypot, including country of origin, IP address reputation, downloaded malware, and so on. Numerical quantities are displayed, such as the number of times the malware was identified by the antivirus software or the number of times the shell command was entered.
- Decision needs to be made quickly, within a few seconds. In addition to being simple to install, the solution must be low performing and able to run on a variety of Linux platforms.

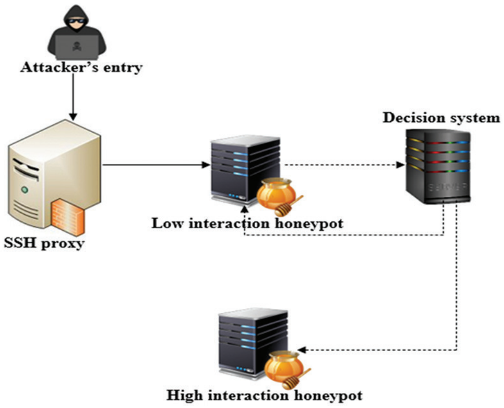

Fig. 4. Schema of the honeynet with a single Low and high interaction honeypot

3.11. BEHAVIOR OF THE ATTACKER

The behavior of an attacker is defined as the sum of all attacks they have carried out in the domain of interest. The attack behaviour h_1 of an attacker a_x in any domain n_y is represented as follows for each attack k_x.

$$h_1 = n_y a_x \Delta k_x \mathrm{a_x} \tag{1}$$

For example, denial-of-service attacks can be carried out on operating systems, databases, hardware, or apps which is shown in Fig. 5.

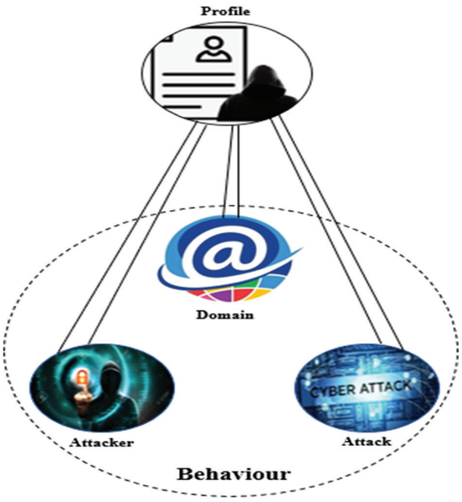

Fig. 5. Relation between profile and behavior of attacker

3.12. PROFILE OF THE ATTACKER

Every a_x will have a behavior, i.e., there will be a behaviour $h_1$for every a_x. A profile F of an attacker indicates the sum of all the attacker's behavior, a_x.$\sum h_1$shows the pattern of a_x' s actions.

$$F = \sum h_1 \tag{2}$$

3.12.1. Properties Of Profile F

Property 1:

Profile F_n is a collection of h_1

Proof:

Set of all possible attacks a_x will define P_m

Set of all possible a_x is a member of any or all h_1

A subset of a_x can be any or all h_1 members.

A specific k_x behavior is specified by a subset of a_x assigned to a particular F_1. Thus, k_x may have only one or multiple h_1. For example, if an attacker sends spam email first, he will be assigned to the behavior h_1. Because the same attacker is responsible for Virus, he has been given the h_2 behaviour. The profile for the attacker is summation of behavior h_1, h_2 which is given in equation 3.

$$Profile\ F = \mathrm{h}_1 + \mathrm{h}_2 \tag{3}$$

So, profile is a group of behaviors.

Property 2:

A cyber attacker cm always performs an action that leads to a purpose, leaving evidence (behaviour) behind. This property was attained by extending Locards exchange principle,

$$\mathrm{h}_1 = \mathrm{c_m k_x} + \sum\ a_x \mathrm{k_x} \tag{4}$$

In contrast, no attack is possible in any domain without a motive involving a set of attack vectors.

3.13. CONTAINER IMPLEMENTATION EFFICIENCY

A container mechanism is not a new concept; the well-known chroot system was first introduced in the 1970s with Unix operating systems, intended to limit the scope of programs. Container implementations such as OpenVZ, Linux Containers, FreeBSD's Jail, and subsequently Rocket and Docker were among the first. In comparison to virtual and physical computers, container instances are extremely light because they lack an operating system and execute just the functions, i.e., services, required for a container's operation. The containers run on the same kernel as the container management system because they are both based on the same operating system. This solution has the following advantages over physical and virtual servers:

- Implementation of infrastructure at a rapid pace
- Minimal footprint
- A high degree of flexibility
- Easy orchestration
- high density

In order to ensure container security, three main mechanisms are used:

3.13.1. Namespaces

The primary and most important security protection for containers is the namespace, as it prevents the containers from learning about host resources, particularly about other container processes or resources that also implement the containers. Docker employs a variety of namespaces in this regard, including User, Net, mnt, and IPC.

3.13.2. CGroups

The Cgroup method ensures that all containers have access to the same resources (CPU, memory, IO). Denial of service attacks that result from bad application behavior or malicious actions occurring in any compromised container are prevented by securing the host or other containers. According to the study's examples, the basic launch of publicly available Docker images generally lacks Cgroup settings.

3.13.3. Capabilities

With the capabilities system, it is easy to control access to containers. Some actions can be performed inside containers, but control actually extends beyond them. In order to avoid the attack footprint as much as possible, the container capabilities are to be minimized to the bare minimum

3.14 FUZZY RULE BASE

The three fuzzy input variables are used to decrease the fuzzy rules. The fuzzy rule base can be applied to reasoning. The security team will be able to predict the possibility of an attack on the low-interaction honeypot by creating this fuzzy rule foundation and fuzzy reasoning engine.

4. RESULTS AND DISCUSSION

The proposed Hybrid H-Doc method was deployed on an AmazonWebService-EC2 instance. These systems do not function as firewalls or IDS/IPS systems, but they provide information on attacks, as well as how to avoid or identify them. Consequently, firewalls and IDS/IPS systems can utilize this knowledge to enhance their capacity to counter such analyzed attacks. Honeypot output can be used to find new threat signatures, blacklist IP addresses, map protocol abnormalities, and so on. The honeypot is a useful analytical and scientific tool as a result.

An innovative hybrid honeynet concept is presented in this study. From the existing methods it is concluded that none of them provide a realistic means to identify the level of complexity of an attack in real time on the basis of its behavior and metadata. A test case scenario is used to evaluate the model's functionality:

The following sections list the outcomes of specific experiments assessed using these criteria.

Table 1. Attacker's information stored in logs

IP address	location	Count of sessions	Severe source	Last seen
59.162.172.25	India, Telangana	532	Yes	2021–10-12 09:46:32
23.25.132.45	India, Maharashtra	846	Yes	2021–10-22 14:54:28
46.206.183.15	Austria, Wien	481	Yes	2021–10-2823:50:14
37.118.125.26	Italy, Marche	654	Yes	2021–10-22 15:44:23

The table below shows the format of the assaults on the honeypot that were logged and kept in the database. A similar table, as represented in Table 2, was recorded for every row in Table 1 to log all the sessions per IP address.

Timestamp	IP address	Sessions	User	Password	Success
2021–10-12 09:46:32	59.162.172.25	45d21 w5423...	Root	#14@es	Yes
2021–10-22 14:54:28	23.25.132.45	124j14 s21a25...	Root	@89dr	Yes
2021–10-2823:50:14	46.206.183.15	541k36 4e12sh..	Root	Gog@13	No
2021–10-22 15:44:23	37.118.125.26	952s21 r236e4...	Root	Kih!26	Yes

Fig. 6 represents the probability of attackers attacking the honeypot system. The honeypot looks like a genuine computer system, complete with apps and data, leading attackers to believe it is a legitimate target. A honeypot gives IT security teams more insight and helps them respond to attacks that the firewall cannot stop.

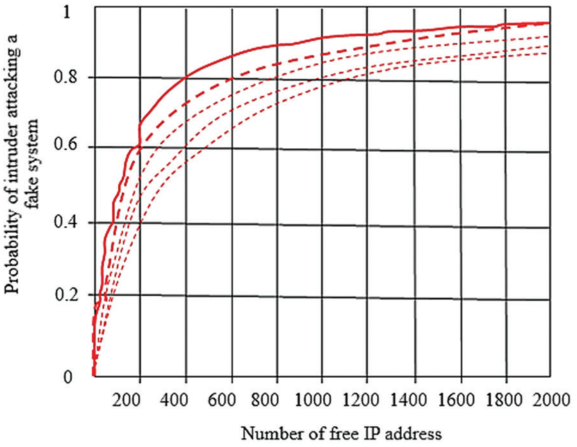

Fig. 6. Probability of intruder attacking Hybrid H-DOC

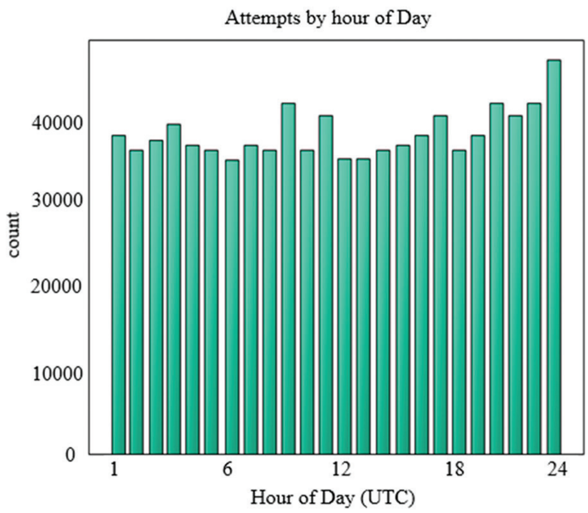

Fig. 7. Count of attacks by attacker in an hour of a day

Fig. 7 represents the count of attacks by attacker in an hour of a day which was recorded in the honeypot.71.14 percent of all cases in the prior year were caused by malware, while 28.86 percent were caused by PUAs. Nearly 86 percent of all spambot occurrences were caused by the Gamut spambot.

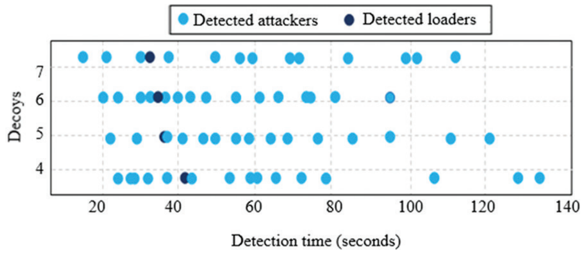

Fig. 8. Effectiveness of decoys

Fig. 8 illustrates how decoys can be useful. A graph showing the recognition of bot attacks and the loader can be seen. There may be four, five, six, or seven decoys in each subnet depending on the circumstances. Using seven decoys per subnet, the loader and attacker's detection time is greatly reduced.

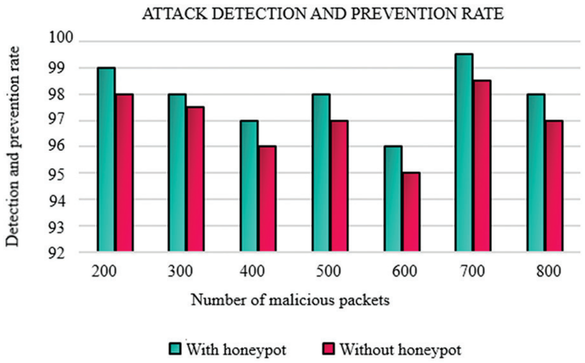

Fig. 9. Rate of attack detection and prevention

The results of the attack detection rate in the two situations are shown in Fig. 9. A performance analysis of attack detection is performed on a total of 800 malicious packets. Results indicate that the suggested framework is more effective when honeypot is implemented.

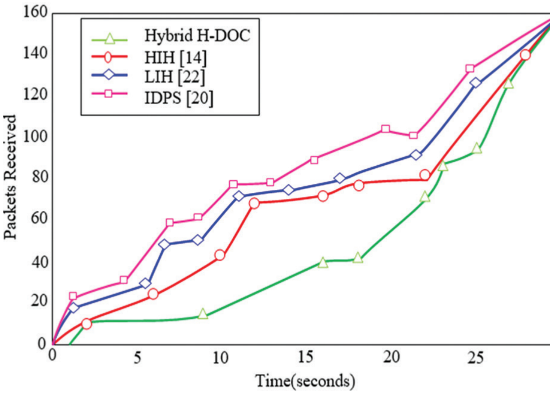

Fig. 10. Comparison of Packet arrival rate

A comparison of the packet arrival times for IDS, LIH, HIH and hybrid H-DOC is shown in Fig. 10. The workload of a hybrid honeypot is lower than that of a standard honeypot, according to the study. As a result, performance of false alarms and workload is improved under all network loads.

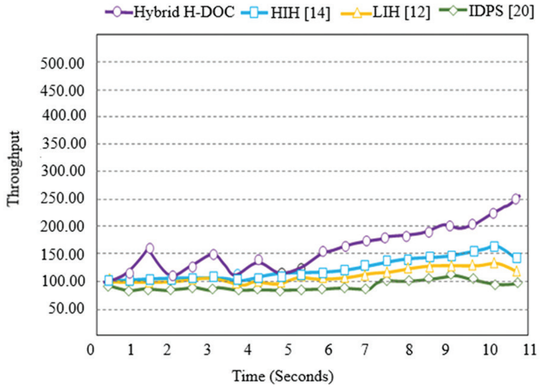

Fig. 11. Throughput of honeypots

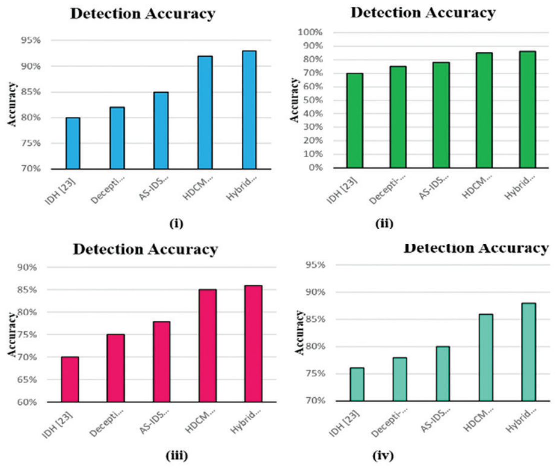

Fig. 12. Comparison of Detection Accuracy (i) When number of attackers=10 (ii) number of attackers=20 (iii) When number of attackers=50 (iv) When number of attackers= 100

The Fig. 12 demonstrates the comparison of detection accuracy of the proposed Hybrid H-DOC bait with the existing methods such as Intrusion Detection Honeypot (IDH), Decepti-SCADA (supervisory control and data acquisition), Anamoly and signature Based IDS (AS-IDS), and Honeypot deployment contract-theoretic model (HDCM). The detection accuracy for the proposed Hybrid H-DOC bait is higher when compared to other existing methods.

5. CONCLUSION

In this paper a hybrid honeynet model has been proposed. Honeypots are set up instantly using Docker technology for practical testing. This system is simple to use, very effective, and capable of recording data from the attacker and capturing malicious attacks. By updating policies, security administrators can enhance their ability to protect the entire application system. The proposed method uses a server with expert system will decide whether the traffic to be given to low interaction systems or too high, so the efficiency is high. The performance evaluation accomplished has demonstrated the feasibility of the proposed solution. We further plan to deploy the honeypot to collect real-world attack data. The collected data will be used for threat intelligence analysis as well as the automated translation of such intelligence into functional cybersecurity configurations, such as rules for firewalls and/or intrusion detection systems.

6. REFERENCES

[1] M. Mirza, M. Usman, R. P. Biuk-Aghai, S. Fong, "A modular approach for implementation of honeypots in cyber security", International Journal of Applied Engineering Research, Vol. 11, No. 8, 2016, pp. 5446-5451.

[2] N. El Kamel, M. Eddabbah, Y. Lmoumen, R. Touahni, "A smart agent design for cyber security based on honeypot and machine learning", Security and Communication Networks, Vol. 2020, 2020.

[3] C. Gupta, "HoneyKube: designing a honeypot using microservices-based architecture", University of Twente, Enschede, Netherlands, Master's thesis, 2021.

[4] E. Chovancová, N. Ádám, "The Security of Heterogeneous Systems based on Cluster High-interaction Hybrid Honeypot", Proceedings of the IEEE 23rd International Conference on Intelligent Engineering Systems, 25-27 April 2019, pp. 81-86.

[5] M. S. Durairajan, R. Saravanan, S. S. Chakkaravarthy, "Low Interaction Honeypot: A Defense

Against Cyber Attacks", Journal of Computational and Theoretical Nanoscience, Vol. 13, No. 8, 2016, pp. 5446-5453.

[6] D. Sever, T. Kišasondi, "Efficiency and security of docker based honeypot systems", Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, 21-25 May 2018, pp. 1167-1173.

[7] R. K. Shrivastava, B. Bashir, C. Hota, "Attack detection and forensics using honeypot in IoT environment", Proceedings of the International Conference on Distributed Computing and Internet Technology, Springer, Cham, 2019, pp. 402-409.

[8] R. Surendiran, "A Secure Command Based Approach to find Stolen Mobiles", Research Review International Journal of Multidisciplinary, Vol. 3, No. 10, 2018, pp. 454-456.

[9] I. M. M. Matin, B. Rahardjo, "Malware detection using honeypot and machine learning", Proceedings of the 7th International Conference on Cyber and IT Service Management, Jakarta, Indonesia, 6-8 November 2019, pp. 1-4.

[10] G. K. Sadasivam, C. Hota, B. Anand, "Detection of severe SSH attacks using honeypot servers and machine learning techniques", Software Networking, Vol. 2018, No. 1, 2018, pp. 79-100.

[11] N. Bhagat, B. Arora, "Intrusion detection using honeypots", Proceedings of the Fifth International Conference on Parallel, Distributed and Grid Computing, Solan, India, 20-22 December 2018, pp. 412-417.

[12] Y. Otoum, A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things", Journal of Network and Systems Management, Vol. 29, No. 3, 2021, pp. 1-26.

[13] G. S. Shiny B. M. Kumar, "E2IA-HWSN: Energy Efficient Dual Intelligent Agents based Data Gathering and Emergency Event Delivery in Heterogeneous WSN Enabled IoT", Wireless Personal Communications, Vol. 122, No. 1, pp. 379-408.

[14] A. Appathurai, R. Sundarasekar, C. Raja, E. J. Alex, C. A. Palagan, A. Nithya, "An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system", Circuits, Systems, and Signal Processing, Vol. 39, No. 2, 2020, pp. 734-756.

[15] N. Innab, E. Alomairy, L. Alsheddi, "Hybrid system between anomaly based detection system and honeypot to detect zero day attack", Proceedings of the 21st Saudi Computer Society National Computer Conference, Riyadh, Saudi Arabia, 25-26 April 2018, pp. 1-5.

[16] S. Suratkar, K. Shah, A. Sood, A. Loya, D. Bisure, U. Patil, F. Kazi, "An adaptive honeypot using q-learning with severity analyzer", Journal of Ambient Intelligence and Humanized Computing, Vol. 13, 2021, pp. 1-12.

[17] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks", IEEE Access, Vol. 8, 2020, pp. 169944-169956.

[18] N. Cifranic, R. A. Hallman, J. Romero-Mariona, B. Souza, T. Calton, G. Coca, "Decepti-SCADA: A cyber deception framework for active defense of networked critical infrastructures", Internet of Things, Vol. 12, 2020, pp. 100320.

[19] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso, L. Armitage, "Cyber threat intelligence from honeypot data using elastic search", Proceedings of the IEEE 32nd International Conference on Advanced Information Networking and Applications, Krakow, Poland, 16-18 May 2018, pp. 900-906.

[20] Y. Yin, Y. Shao, X. Wang, Q. Su, "A Flexible Cyber Security Experimentation Platform Architecture Based on Docker", Proceedings of the IEEE 19th International Conference on Software Quality, Reliability and Security Companion, Sofia, Bulgaria, 22-26 July 2019, pp. 413-420.

[21] J. Buzzio-Garcia, "Creation of a High-Interaction Honeypot System based-on Docker containers", Proceedings of the Fifth World Conference on Smart Trends in Systems Security and Sustainability, London, United Kingdom, 29-30 July 2021, pp. 146-151.

[22] S. Sivamohan, S. S. Sridhar, S. Krishnaveni, "Efficient Multi-platform Honeypot for Capturing Real-time Cyber Attacks", Intelligent Data Communication Technologies and Internet of Things, Springer, Singapore, 2022, pp. 291-308.