© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 03 | NCASIT- 2022 | April 18th 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed

Hybrid honey pot system for malware analysis using python

Ritesh Kawale
Computer Engineering Student
Saint Vincent Pallotti College Of Engineering and Technology
Nagpur, India
Ritesh.kawale99@gmail.com

Aditya Dhumal
Computer Engineering Student
Saint Vincent Pallotti College Of Engineering and Technology
Nagpur, India
adityad.ce17@stvincentngp.edu.in

Aditya Nipane
Computer Engineering Student
Saint Vincent Pallotti College Of Engineering and Technology
Nagpur, India
adityan.ce18@stvincentngp.edu.in

Tanvi Mankar
Computer Engineering Student
Saint Vincent Pallotti College Of Engineering and Technology
Nagpur, India
Tanvim.ce16@stvincentngp.edu.in

Abstract- **The latest wireless technology is growing smartphone technology and emerging mobile cloud technology. Mobile cloud computing has a lot of advantages in the future, but it's also very easy for hackers to take full control of the privacy of many other users' data. While data security is expected to be secure, the main disadvantage for users is that when the computer is connected to the internet, an intruder can easily steal data from the required target. As a result, a combination of Hybrid Intrusion Detection System (HyInt) and Honeypot networks have been implemented into the Mobile Cloud Environment to provide better security by mitigating unidentified and known attacks. The research work's execution provides a pure perspective of the algorithm's security and quality products that were not included in the previous research work. Intensive statistical analysis was carried out as part of the research to demonstrate the consistency of the proposed algorithm. The implementation and evaluation results show that there is plenty of room for more research on the cloud-based Intrusion Detection System. The implemented algorithm can be used to effectively monitor the network's activities in a high-security cloud environment developed for army and banking purposes.**

Keywords: Performance, Hybrid Intrusion Detection System, Signature and Anomaly-based detection, Honeypot Networks, Mobile Cloud Computing.

I. INTRODUCTION

A honeypot can be used in network security to discover new attacks that Intrusion Detection Systems or network firewalls may not be able to detect using the old static defense rule system. When designing IDS (Intrusion Detection Systems) and Firewalls, it is critical to consider the enterprise defense rules for going through the honeypot. Computer networks are vulnerable to a variety of exploits that can render them insecure or prevent them from performing their intended function. Intruders and attackers have become increasingly agitated about network security and challenges. Enterprises, organizations, and, more importantly, finance departments have an essay solution to implement various hardware and software for network security providers such as firewalls, variants of the intrusion detector[18], and Virtual Private Networks to have a better and improved security. These solutions, on the other hand, work nonstop to keep proprietary information out of the hands of unscrupulous intruders, and to warn of new attacks as they occur.

Mobile Virtualization is the most advanced feature emerging in today's world, and its applications for smartphones are growing by the day. The number of mobile users is growing all the time because it makes work easier and faster, as well as provides the latest technology that is rapidly evolving and allows users to access all of their apps via the network from anywhere in the world. Mobile cloud computing has a major advantage in that it is very versatile, and we can access data and share information anywhere in the world even if we are not connected to the internet. It also offers cost-effectiveness in that use and maintenance are comparatively low, and real-time data availability, where all user information is available in real-time on our mobile device when connected to the network, from which we can update and acquaint ourselves. Despite all of the hype surrounding the MCC, it has a major flaw in terms of privacy and security, which contributes to trust issues for consumers and businesses. As the world evolves, so does the number of hackers.

Similarly, companies are implementing new things and methods for protection in the cloud computing environment, where cloud computing services are available on a pay-as-you-go basis. When used together, how do the Hybrid Intrusion Detection System (HyINT) and Multi-Honeypot Network (MHN) improve security in Mobile Cloud Computing? Honeypot networks are used to achieve more defense-in-depth protection and total security of the cloud environment. The analysis of attack approaches is identified in the honeypot networks network as necessary for countermeasures. Many harmful threats, such as DDOS, XSS injection, and SQL injection, can't be completely avoided, but they can be minimized. There are several ways to protect it from hackers, but IDS is the most important and widely used method for detecting malicious code in a network, and it plays a critical role in protecting the cloud environment from attackers.

II. LITERATURE SURVEY

The following papers were studied which were relevant to agriculture and inference was drawn from each is mentioned below.

Sr.No	Detail of Paper		Problem Identification	Paper Approached for the Problem	Result/Dataset
1.	DamienWarren Fernando, et. al," A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques"[28]		The main motivations for this study are the destructive nature of ransomware, the difficulty of reversing a ransomware infection, and how important it is to detect it before infecting a system.	The exploration into machine learning and deep learning approaches when it comes to detecting ransomware poses high interest because machine learning and deep learning can detect zero-day threats. These techniques can generate	We carried out experiments to investigate how the discussed research studies are impacted by malware evolution. We also explored the new directions of ransomware and how we expect it to evolve in the coming years, such as

				predictive models that can learn the behavior of ransomware and use this knowledge to detect variants and families which have not yet been seen. In this survey, we review prominent research studies which all showcase a machine learning or deep learning approach when detecting ransomware malware.	expansion into IoT (Internet of Things), with IoT being integrated more into infrastructures and into homes.
2.	Umara Urooj, et. al," Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions " [29]		Ransomware is an ill-famed malware that has received recognition because of its lethal and irrevocable effects on its victims. The irreparable loss caused due to ransomware requires the	This study provides information about the collection of the dataset from its sources, which were utilized in the ransomware detection studies of the diverse platforms. This study is also distinct in terms of providing a	This survey is intended to provide a user manual that can encourage researchers as a direction to work with available technologies in the field of ransomware

			timely detection of these attacks.	survey about the ransomware detection studies utilizing machine learning, deep learning, and a blend of both techniques while capitalizing on the advantages of dynamic analysis for ransomware detection. The presented work considers the ransomware detection studies conducted from 2019 to 2021. This study provides an ample list of future directions which will pave the way for future research.	attack detection. It can help them in developing more efficient ransomware detection models while considering the available solutions. In the future, we shall work on the significance and contribution of static analysis for the detection of ransomware attacks utilizing machine and deep learning methods.
3.	Craig Beaman, et. al," Ransomware: Recent advances, analysis, challenges and future research directions"[30]		The COVID-19 pandemic has witnessed a huge surge in the number of ransomware attacks. Different institutions such as healthcare, financial, and	In this work, recent advances in ransomware analysis, detection, and prevention was explored. It was found that the focus	Through the experiments, it was also observed that ransomware can be easily created and used. In the end, we highlighted the

			government have been targeted. There can be numerous reasons for such a sudden rise in attacks, but it appears working remotely in home-based environments could be one of the reasons.	of the state-of-the-art ransomware detection techniques mostly revolve around honeypots, network traffic analysis, and machine learning-based approaches.	existing research challenges and enumerated some future research directions in the field of ransomware.
4.	Faizan Ullah," Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls"[31]		Ransomware (RW) is a distinctive variety of malware that encrypts the files or locks the user's system by keeping and taking their files hostage, which leads to huge financial losses to users.	The proposed model can detect a large number of RW from various families at runtime and scan the network, registry activities, and file system throughout the execution. API-call series was reutilized to represent the behavior-based features of RW. *e technique extracts fourteen-feature vector at runtime and	To validate the effectiveness and scalability, we test 78550 recent malign and benign RW and compare with the random forest and AdaBoost, and the testing accuracy is extended to 99.56%.

				analyzes it by applying online machine learning algorithms to predict the RW.	
5.	Samuel Egunjobi," Classifying Ransomware Using Machine Learning Algorithms"[32]		Detection and mitigation systems have been developed and are in wide-scale use; however, their reactive nature has resulted in a continuing evolution and updating process.	In this paper, we demonstrate a classification technique of integrating both static and dynamic features to increase the accuracy of detection and classification of ransomware. We train supervised machine learning algorithms using a test set and use a confusion matrix to observe accuracy, enabling a systematic comparison of each algorithm.	In this work, supervised algorithms such as the Na• ve Bayes algorithm resulted in an accuracy of 96% with the test set result, SVM 99.5%, random forest 99.5%, and 96%. We also use the Youdens index to determine sensitivity and specificity.

.

III. SYSTEM ARCHITECTURE

A. Worms Activity

In the context of a network, a worm is a piece of software or a program that, when run on a honeypot, causes other honeypots to modify their administration to the point where they begin to form links and generate connection or pair

connection requests. This delimitation aids in the identification of non-self-distributing network actions from self-distributing network actions that take the system down and configure it according to its code. It does not, however, intend to continue the method automatically. Almost all types of worms have their executable code, indicating that the captured worms have multiple links and may have experienced a system buffer overflow or password generation. Even though most of these viable or executables have a nickname that is most directly associated with them, and because they are initially available as files by the worms, The following Table I. shows the various worm models as well as the number of worms captured on our network.

The proposed work provides the best architecture that focuses on the best decoy and lure architectures that are absorbed by internal network attacks via a hybrid honeypot that can capture and record all incoming and existing data and provide us with data control. The proposed honeynet records all intruder activities and operations and sends them to a log for further analysis.

B. Data Analyzing Module

The data analyzing module examines the information gathered from the original data. The honeynet collects data from internal honeypots and sends it to be analyzed. In the meantime, we're using an appropriate firewall to get more information about the data we've captured, and we're also sending the firewall logs to our analyzer. In the proposed architecture, a firewall module will act as a logger, capturing all traffic and its status in our back-end design, allowing us to access our production systems.

C. Honeynet Activity

As previously stated, the honeynet performs two primary functions: information control and information seizure or data recording. The primary goal of information control is to prevent intruders from using the honeynet feature to gain access to the other host. The goal of information seizure is to capture all of the functionality of intruders. It is difficult to gather information as quietly as possible while remaining undetected by intruders. Most intruders attempt to spread through encipher channels such as SSL (Secure Sockets Layer), IPSec, SSH (Secure Shell), and other related channels.

Encryption must be performed with a specific account by the data collector mechanism in such activities. In addition to this, we use seizer tools with similar functionality on the honeypot to achieve a multi-record level method of recording [1]. This way, not only can you connect the various intruders' activity steps, but you can also keep the path away from the default of a single mechanism.

Logs, information recorded, and system activity recorded by honeypot tools are transferred to the analyzing module. The data is saved as obtain information consistent with the network connection feature and its contents. The honeynet recorded information has a smaller amount size, but it is more fidelity and fatal.

We can set up a virtual honeypot [14] on a host by taking advantage of virtual technology, which is also used in honeynet. This strategy assists in deducting and minimizing the cost of honeynet development. Nonetheless, the performance required to deploy a host is still higher.

D. De-Militarized Zone

Because a De-Militarized Zone (DMZ) is not a network hardware device like a router or a bridge [8,] it does not pass through different packets. The De-Militarized-Zone (DMZ) is intended to provide secure communication with servers before packets enter a firewall, without the need for any inbound firewall gaps between the internal LAN or network and the deployed DMZ.

The policy specifies the data security requirements for networks, as well as the machines and peripherals used within the DMZ. Traditional De-Militarized Zones allow machines located behind a firewall to comment on requests destined for the DMZ. Machines in the DMZ respond to, attempt to forward or reissue queries sent from outside the internet or public network.

Many DMZs use a server (such as a proxy server) or other servers as the machines deployed within the DMZ. The firewall was installed after attempting to prevent machines in the DMZ from initiating inbound requests. For the DMZ configuration, the majority of the machines on the internal network or in a typical LAN run behind the firewall, allowing them to connect to an external network or the internet. To deploy the secure zone, a few machines or servers are also used outside the firewall in the DMZ; these machines intercept traffic and agent queries for other parts of the network, and they provide an extra layer of protection for the machines behind the firewall zone.

A DMZ typically includes servers that provide various internet-based services to clients. These services include FTP, e-mail services such as SMTP, IMAP4, and POP3, and a DNS server. Even though these servers must have direct internet access, they can also protect the firewall. The servers and honeypots could be located in the DMZ or inside the network, but the DMZ is recommended. The best structure that we are looking for is shown in Fig. 1.

E. Proposed Hybrid Honeypot Framework

The proposed advancement introduces an adaptable honeypot-based network security system that has been adopted to change, in particular, organizational, financial, and critical conducted server zone networks based on the energetic dynamic implementation and configuration of hybrid honeypots.

The primary idea behind low interaction honeypots is to use free, ready-to-use unused IP addresses made available by operating systems or distributed ones and their services. They simulate the distributed operating systems and services of deployed production hosts in a specific network. In most cases, going network traffic to honeyds will be directed to high interaction honeypots where attackers will interact with specific services. The use of half-breed or hybrid animals to approach honeypot technology falls into two categories:

Using the least amount of administrative interference due to the number of honeyds and their specific service setups automatically based on network authority. Focusing on the need for honeynets or high interaction honeypots in the network through traffic redirection from low interaction honeypots demonstrates the affection of honyds as real systems to attackers.

F. Proposed Honeynet

Because of the presence of fake machines in the network, the system administrator must first assign the IP addresses of the physical honeypots or essential hosts in the honeynet, then authorize traffic redirection from low interaction honeypots, and log the activities of attackers. The locution redirection does not simply change the communication direction between machines. It was, however, about reformatting the entering network packets predetermined to a specific honeyd and returning them to the network. They will be able to discover their way to the true honeypot if they deploy in this manner. Following that, he responds to the intruder and then convinces him that the invader is interacting with a real machine.

For our hybrid honeypot approach, we try to show an example of typical Local Area Network behavior. Figure 1 depicts the deployment as well as the position. This diagram depicts a low interaction honeypot server that is directly connected to the main switch and other production systems. It also shows the physical honeypots in the architecture honeynet that are ready to receive network direct traffic or traffic that has been redirected through a low interaction honeypot. As shown in the architecture, the low interaction honeypots machines appear to be physical or production systems, but in reality, they are just advanced virtual machines.

We may employ Network Address Translation in our architecture (NAT). This method avoids the need to reconfigure each honeypot to be dynamically in an internal domicile for external domiciles that arrive via NTM (Network Traffic Monitoring). As a result, we should point out that by configuring the honeypot to support dynamic address reconfiguration, we can avoid this step entirely.

The low interaction honeypot server depicted in this figure has three main functionalities that imply different threads. The first honeyd server interfaces with a network scanning application to gather information about the various operating systems available in the network, their specific direct or administer ports, and their running services, and then collects and saves this information in a file. The following thread reads the data from the file and adjusts the required configuration of low interaction honeypots. As a result, it includes the operating system, their services, and the distribution of port and network assistance in the real network part. The final thread examines the low interaction honeypot log traffic data and saves it to a specific file. Furthermore, while invaders are active, the servers wait for arriving traffic that is directed to unused IP addresses and then presumes to identify those IPs.

To build the proposed system depicted in Fig. 1, a programming language, network scanning tools, and operating system must be selected. Even though the approach architecture framework, in general, is advanced and not limited to a specific preference. Because of its flexibility in deploying the security application, the operating system chosen for the honeyd server required open source. We used the Linux Fedora 12.0 version for this purpose because it has the required feasibility due to our framework. The programming language required network library availability language functionality as well as the ability to simply integrate Fedora tools. In such cases, we choose Python, which provides us with the required library of available networking.

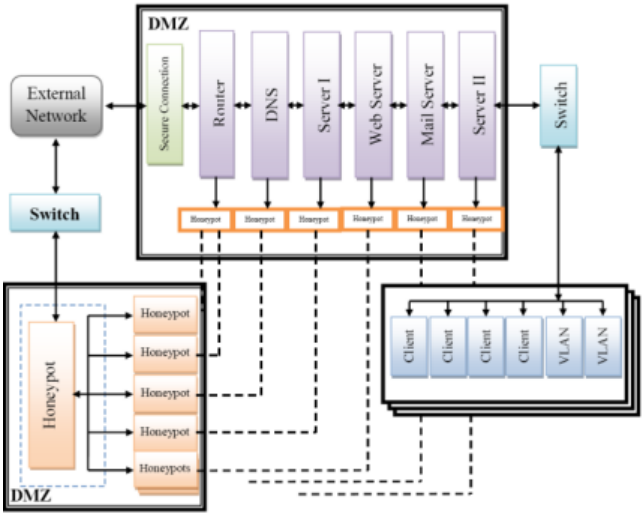

Figure 4. Hybrid honeypot architecture.

Following that, a network scanning tool was required to resolve the type of operating system used in the network as well as the various ports in the production network and provide such required information.

The Nmap was chosen for a specific purpose as part of our experiment. This network tool can be used in two distinct active modes to collect data about various available distributed operating systems, as well as to conduct ports and assumed network operating services. These two folded are normal modes, in which this tool gathered the information at the precise time. In this mode, Nmap tries to parallelize port scans; however, while information can be collected in a short period, the server may become overloaded with input or output data, causing network traffic to increase accordingly. The polite mode of the Nmap tool gathers information in a time-consuming manner. The tool serializes port scans while hesitating between sequential scans in this mode. This case applies to the machine and the network being amicable in terms of time consumption and taking a long time to complete the scans. Nonetheless, we will have a thorough scan of the network.

As shown in Fig. 2, the process of Nmap scans consists of sending a ping to establish all devices on the network and collect their IP addresses, but not permanently in a specific file. This file can be used to perform the next scan, which will be operating system or port scans for the given IP addresses. The results of the scans are logged into a specific file, which is generally now a day using the property of an XML file that is analyzed every time until the scan is completed. When the tool scans are finished and stopped, an analyzer begins and runs in a thread to extract the collected data from the file, which automatically creates a profile to store these data.

G. Deployment of Honeyds

The primary arguments for proposing the hybrid honeypot are to make use of unused IP addresses; however, there is a task that helps to solve how to separate them among the running operating system and thus minimize the likelihood of revealing the real and production hosts in the network and allowing them to be attacked by intruders. A straightforward advance was deployed to ensure a constant continuation after integrating the virtual systems into the production system via operating system distribution, and it should be while extinguishing the physical honeypots.

.

IV. IMPLEMENTATION DETAILS

In this section, we provide background information as well as work related to the proposed solution in this paper. Where we talk about Mobile Cloud Computing and its security concerns. 2.1 Mobile Cloud Computing as a Focus Mobile Cloud Computing is a current and trending technology all over the world, and it has a variety of advantages that are very useful in terms of enriching the user experience. [2] Its specific functions include storage and smartphone mobility anywhere via wireless or internet access, and its service is simply paid as you go. Similarly, as a result of Juniper Research, the growing use of mobile computing, which notes that the public and private sector demand for cloud-based mobile applications is expected to increase to 9.5 billion dollars by 2014, but hopefully more than that shortly. Similarly, smartphone applications have become numerous in recent years, with applications in various categories such as entertainment, social media, online streaming, banking, news, and so on. The main reason for this is that mobile computing is capable of providing the subscriber with a resource where and how it is required purely based on the user organization. According to a 2009 International Data Corporation (IDC) study, 74 percent of IT administrators and Chief Information Officers (CIOs) believe that user privacy concerns are the major risks that have kept most organizations from adopting virtualization. Three fundamental principles benefit mobile computing the most: technology, hardware, and communications. Whereas hardware consists of devices such as smartphones and portable devices that clients can use. However, with the rapid advancement of wireless networks, consumers are gradually adopting PDAs. [3] According to the Allied Business Intelligence report, more than 2.4 billion consumers will use a portable device to access a cloud computing platform in 2015. Similarly, Google highlights certain cloud-based products for consumers and businesses, where it has a necessary item for mobile phones that are currently trending all over the world known as Android OS, as well as various applications such as Google Maps, Street View, and so on. Similarly, Google has launched Google Stadia, a cloud-based gaming service that does not require any hardware and only requires an internet connection to connect. [4] The design of the MCC process is depicted in Figure 1. The core techniques used in the technology industry, such as parallelization, virtualization, and mass production, are the three primary techniques for cloud computing.

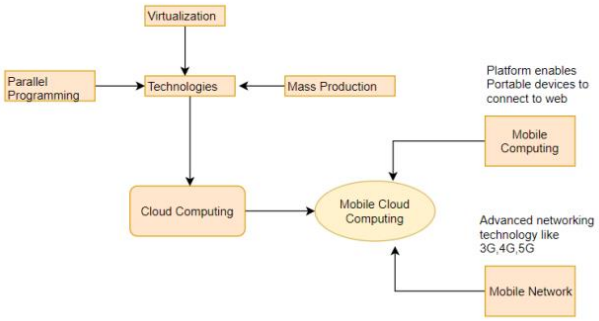

Figure 1: Architecture of Mobile Cloud Computing

Importance of Cloud Security

As people use mobile phones in cloud environments, mobile devices are vulnerable to a variety of external threats that can result in unknown attacks, whereas information privacy and authentication should be known to regular users and software developers, as if they are aware of the outcomes of the privacy, there will be no problems with hackers. People nowadays are unaware of how to use technology and the advanced features on their smartphones and PDAs. Mobile protection can be obtained through a variety of security features, including app installation, such as anti-virus software. [1] [5] Formalized paraphrase Mobile Cloud Computing (MCC) security frameworks are divided into two categories: application security and data security frameworks. Storing data on a database in a virtual environment without revealing any details is more difficult for mobile users. An authentication method is used to verify that if a user transfers a file to a cloud server for sharing with different clients, it should also be checked that the user accessing the file is a trustworthy client. Scalability is the ability of a network to interact with clients flawlessly. [19] Similarly, the latest security technologies for online services, such as VPN usage, password encryption, authentication, and entry command, should be introduced to provide uninterrupted services against various attacks, such as DOS attacks and data theft. [23] As a result, when such attacks occur, cloud

services must provide a backup and restore service to boost customer trust. Figure 2 depicts recent security issues and current approaches in the table below.

Security Issues		Current Approaches
Mobile Cloud	Platform Reliability	Authentication and access control, Privacy and data protection.
	Privacy and Data Protection	Key management and data encryption. Integrating the current security technologies.
Mobile Terminal and Network	Malware software	Detection and prevention CloudAV
	Software Vulnerabilities	Installing the system patches, checking software legitimacy and integrity.
	Information Leakage	Data Encryption and Security Protocol

Figure 2: List of Security Issues

Potential of Intrusion Detection

Cloud-based system An intrusion is an attack that may compromise the CIA of a device or network, and there are numerous types of intruder attacks. (DOS) attacks are the most common. Denial of Service occurs when legitimate users are unable to access internet-based services. [6] In the virtual environment, the intruder can send repeated attempts to authenticate VMs using cyborgs, causing their availability to legitimate users to be overburdened. The implementation of still-accessible Intrusion Detection and Prevention Systems (ID/ PS) could not achieve the required level of protection and performance. Pandeeswari and Kumar (2016) used a Fuzzy Mean Clustering-based ANN to detect breaches in the cloud, where IDS is typically implemented on end-host cloud servers. [5] [7] Potential ransomware will prevent the use of traditional HIDS based on signature matching methods by employing authentication techniques. Complex evaluation based on existing IDS can be avoided by testing the controlled computer with the help of the security process. [8] Signature matching approaches necessitate proper monitoring, and a subsequent level of protection (Modi and Patel, 2013) connects modern NIDS tools with traditional anomaly detection methods to detect cyberattacks on a network. Similarly, some cloud-protected services, such as Snort IDS, fail to recognize VM attacks aimed at different residents on a physical server. Figure 3 depicts the various types of Cloud IDS. [9] The effectiveness of IDS can be significantly increased by combining signature-based techniques with anomaly-based techniques in the Hybrid Intrusion Detection System.

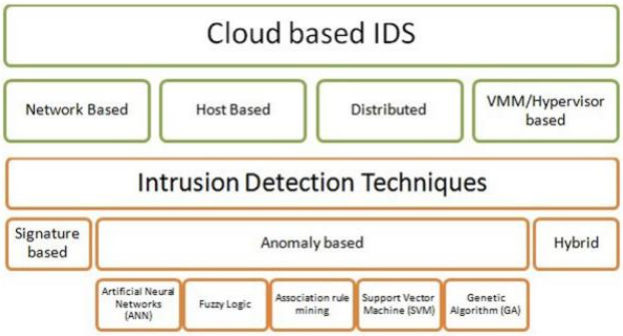

Figure 3: Structure of Intrusion Detection System in Cloud environment

The ability to withstand new unknown attacks by leveraging existing knowledge generated by known attacks. X. Wang et al proposed a methodology related to the central management approach, despite having the drawbacks of all strategies that use centralized control in a distributed environment [10]. Similarly, Modi et al. pioneered a method for stepwise intrusion detection. It initially pre-processes packets and sends them to signature-based IDS after comparing them to previously discovered patterns. Hybrid IDS is more beneficial in terms of vulnerability security and performance. Previous solutions' main limitations were that they could not be fully designed to handle new types of attacks, which is also a time-consuming task that requires far too much time to investigate suspicious attacks.

Honeypot uses Intelligence

Honeypots are a sophisticated idea in network security. Such a system aims to gather information about intrusion attempts. The level of interaction ranges from minimal interaction honeypots that only simulate the communication layer to strong interaction honeypots that run a real operating system. One of the primary reasons for using cloud services is to benefit from lower IT infrastructure and company costs, and it is to collect high and low communication honeypots used in a cloud environment to evaluate attacks, they must verify that the distributed packets are legitimate once they are transferred to HoneyCY as their transition to the cloud [12]. Similarly, it is composed of three design layers, where HoneySrv collects honeypie devices and information gathered, and HoneyVm analyses malware collected. Brown et al listed several virtualization systems involved in honeypot sensors, and Saadi et al provided an IDS focused on a smartphone device with a mix of honeypots such as Honeycomb, HoneyNet, and HoneyD. [13] [14] Formalized paraphrase.

V. RESULT

On our dataset, we used machine learning algorithms such as Random Forest Classifier, Decision Tree Algorithm, Support Vector Machine, Gaussian Naive Bayes Classifier, and Logistic Regression, and the accuracy obtained by these algorithms is as follows.

Ransomware Detection using HoneyPot Machine learning

Ransomware is a type of malware that encrypts the files of a victim. The attacker then demands a ransom from the victim in exchange for restoring access to the data. There are several ways ransomware can gain access to a computer. One of the most common methods of delivery is phishing spam, which consists of attachments sent to the victim in the form of an email disguised as a file they should trust. They can take over the victim's computer once they've been downloaded and opened, especially if they have built-in social engineering tools that trick users into granting administrative access. Other, more aggressive forms of ransomware, such as NotPetya, use security flaws to infect computers without the need to trick users. There are several methods by which attackers select which organizations to target with ransomware. Sometimes it's a matter of chance: for example, attackers may target universities because they have smaller security teams and a diverse user base that engages in a lot of file sharing, making it easier to breach their defenses.

A honeypot is a network-connected system that is used as a trap for cyber-attackers to detect and study the tricks and types of attacks used by hackers. It simulates a potential target on the internet and alerts the defenders to any unauthorized attempt to access the information system. Honeypots are mostly used by large corporations and cybersecurity organizations. It assists cybersecurity researchers in learning about the various types of attacks used by attackers. It is suspected that cybercriminals also use honeypots to deceive researchers and spread false information.

Result

R2L

```
In [54]: accuracy = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='accuracy')
         print("Accuracy of Ransomware Data Set: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))
         precision = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='precision_macro')
         print("Precision of Ransomware Data Set: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))
         recall = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='recall_macro')
         print("Recall of Ransomware Data Set: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))
         f = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='f1_macro')
         print("F-measure of Ransomware Data Set: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))

         Accuracy of Ransomware Data Set: 0.96705 (+/- 0.00752)
         Precision of Ransomware Data Set: 0.95265 (+/- 0.01248)
         Recall of Ransomware Data Set: 0.95439 (+/- 0.01401)
         F-measure of Ransomware Data Set: 0.95344 (+/- 0.01070)
```

Figure 5. Ransomeware Detection

Remote-to-user (R2L) attacks are a type of computer network attack in which an intruder sends a series of packets to another computer or server over a network that he or she does not have permission to access as a local user.

Classification when we use the term accuracy, we usually mean precision. It is the number of correct predictions divided by the total number of input samples.

$$\text{Accuracy}=\frac{Number\ of\ Correction\ predictions}{by\ the\ total\ number\ of\ the\ prediction\ made}$$

The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp is the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample.

A. DOS Attack Detection using HoneyPot Machine learning

A Denial-of-Service (DoS) attack attempts to bring a machine or network to a halt, rendering it inaccessible to its intended users. DoS attacks achieve this by flooding the target with traffic or sending it information that causes it to crash. In both cases, the DoS attack deprives legitimate users (employees, members, or account holders) of the service or resource they anticipated.

DoS attacks frequently target high-profile organizations' web servers, such as banks, commerce, and media companies, as well as government and trade organizations. Though DoS attacks do not usually result in the theft or loss of valuable information or assets, they can cost the victim a significant amount of time and money to deal with.

DoS attacks can be classified into two types: flooding services and crashing services. Flood attacks happen when the system receives too much traffic for the server to buffer, causing it to slow down and eventually stop working.

Result

```
In [31]:  from sklearn.ensemble import RandomForestClassifier
          rfc = RandomForestClassifier()

          #fit random forest classifier on the training set
          rfc.fit(train_x, train_y)

          # extract important feature
          score = np.round(rfc.feature_importances_, 3)
          importances = pd.DataFrame({'feature': train_x.columns, 'importance':score})

          importances = importances.sort_values('importance', ascending=False).set_index('feature')
          print("==================================Fature of DOS Attack==================================")
          # plot importances
          plt.rcParams['figure.figsize'] = (11,4)
          importances.plot.bar()

          ==================================Fature of DOS Attack==================================

Out[31]:  <AxesSubplot:xlabel='feature'>
```

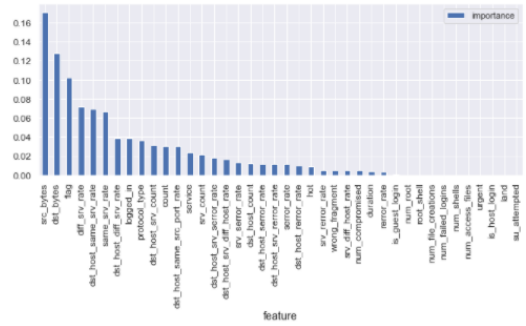

Figure 6. DOS Attack Detection

A random forest is a meta estimator that uses averaging to improve predictive accuracy and control over-fitting by fitting several decision tree classifiers on different sub-samples of the dataset. If bootstrap=True (the default), the sub-sample size is controlled by the max samples parameter; otherwise, the entire dataset is used to build each tree.

Random Forest Classifier is implemented using Python's Scikit-Learn library and the IRIS dataset, which is a well-known and widely used dataset. The Random Forest, also known as the Random Decision Forest, is a supervised machine learning algorithm that uses decision trees to perform classification, regression, and other tasks.

The Random forest classifier generates a set of decision trees from a subset of the training set that is chosen at random. It is essentially a set of decision trees (DT) drawn from a randomly selected subset of the training set, and it then aggregates the votes from the various decision trees to determine the final prediction.

VI. CONCLUSION

The proposed hybrid honeypot architecture system protects production systems to some extent. It achieves this by lowering the hacker's likelihood of activity and targeting our production systems through the use of lure systems in the network, where the hacker cannot learn about these systems, their status, or his fingerprint and instead considers the fake systems to be real systems. This will not achieve our goal unless we use the redirection capability, and the production system will remain vulnerable to direct attacks that do not pass through the conducted honeypot system. In the proposed design, the production honeypots can only play a passive role, logging different activities of the attackers so that the system administrator can extract and analyze them using data mining. This could play a more active role by analyzing the attacker's activities and reducing the different types of attacks through the use of a signature file or a signature database that has the capability of development and data mining. As previously

demonstrated, honeypots will be capable of adding and releasing warnings, as well as sending notifications to the administrator, the type of intruder, and various feasible suggestions to prevent attack propagation.

VII. ACKNOWLEDGEMENT

We would like to express our very great appreciation to Cojag Smart Technology Pvt Ltd for their valuable and constructive suggestions during the planning and development of this research work. Their willingness to give us time so generously has been very much appreciated.

We would also like to thank the staff of the St.Vincent Pallotti College of Engineering & Technology for enabling us to visit their computer labs.

VIII. REFERENCES

[1] Camilo, Viecco. “Improving Honeynet Data Analysis,” Information Assurance and Security Workshop, pp. 99-106, 2007.

[2] D. Moore, “Network telescopes: Observing small or distant security events,” Proceedings of the 11th USENIX security symposium, 2002.

[3] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Network telescopes: Technical report," CAIDA, April 2004.

[4] Dacier M, Pouget F, Debar H. Honeypots: practical means to validate malicious fault assumptions. In: Proceedings of 10th pacific rim international symposium on dependable computing, pp. 383–8, March 2004.

[5] Eugene Spafford. An analysis of the Internet worm. In Proceedings of European Software Engineering Conference, September 1989.

[6] Evan Cooke, Michael Bailey, Z. Morley Mao, David Watson, Farnam Jahanian, and Danny McPherson. Toward understanding distributed blackhole placement. In Proceedings of the Second ACM Workshop on Rapid malcode (WORM), October 2004.

[7] http://www.pandasecurity.com.

[8] http://www.sans.com.

[9] J. Dike, “User-mode Linux,” Proceedings of the 5th annual conference on Linux Showcase & Conference-Volume 5, USENIX Association Berkeley, CA, USA, pp. 2-2, 2001.

[10] Khattab M, Sangpachatanaruk C, Mosse D, MelhemR, Znati T. Roaming honeypots for mitigating service-level denial-of-service attacks. In: Proceedings of the IEEE 24th international conference on distributed computing systemsMarch, p. 328–37, 2004.

[11] Krawetz N. Anti-honeypot technology. IEEE Security & Privacy Magazine, Vol. 2(1), pp. 76–9, 2004.

[12] Kreibich C, Crowcroft J. Honeycomb: creating intrusion detection signatures using honeypots. ACM SIGCOMM Computer Communication Review, Vol. 34(1), pp. 51–6, 2004.

[13] Kuwatly I, Sraj M, Al-Masri Z, Artail H. A dynamic honeypot design for intrusion detection. In: Proceedings of IEEE/ACS international conference on pervasive services, p. 95–104, July 2004. [14] Lok Kwong Yan. “Virtual honeynets revisited,” Information Assurance Workshop, pp 232-239, 2005.

[15] Mark Eichin and Jon A. Rochlis. With microscope and tweezers: An analysis of the Internet virus of November 1988. In Proceedings of the 1989 IEEE Symposium on Security and Privacy, 1989.

[16] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Scalability, _delity, and containment in the Potemkin virtual honeyfarm. In Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP), October 2005.

[17] Omid Mahdi Ebadati E., Harleen Kaur and M. Afshar Alam."A Performance Analysis of Chasing Intruders by Implementing Mobile Agents". International Journal of Security (IJS), Vol. 4, No. 4, pp 38-45, 2010.

[18] Omid Mahdi Ebadati E., Kaur H., Alam A.M., "A Secure Confidence Routing Mechanism Using Network-based Intrusion Detection Systems", OLS Journal of Wireless Information Networks & Business Information System, Open Learning Society, Nepal, pp. 83 – 93, 2010.

[19] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," ACM SIGOPS Operating Systems Review, vol. 37, pp. 164-177, 2003.

[20] Spitzner L. Honeypots: tracking hackers. AddisonWesley,; 2002.

[21] Teo L, Sun A, AhnJ. Defeating internet attacks using risk awareness and active honeypots. In: Proceedings of the second IEEE international information assurance workshop, p.p. 155–67, April 2004. Virtual PC, http://www.microsoft.com/windows/products/winfamily/virtualpc/ default.mspx, 2008.

[22] Virtualbox, http://www.virtualbox.org, 2008.

[23] VMWare, http://www.vmware.com, 2008.

[24] W32.Blaster.Worm. http://securityresponse.symantec.com/avcenter/venc/data/w32.blast er. worm.html.

[25] Weiler N. Honeypots for distributed denial of service attacks. In: Proceedings of the 11th IEEE international workshop on enabling technologies: infrastructure for collaborative enterprises (WETICE'02) June 2002.

[26] Wolfgang M. Host discovery with Nmap, 2002.

[27] Yeldi S., Gupta S., Ganacharya T., Doshi S., Bahirat D., Ingle R., et-al. Enhancing network intrusion detection system with honeypot. Conference on Convergent Technologies for Asia-Pacific Region TENCON 2003, p. 1521–6, October 2003.

[28] DamienWarren Fernando , Nikos Komninos and Thomas Chen," A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques", IoT 2020, 1, 551–604; doi:10.3390/iot1020030

[29] Urooj, U.; Al-rimy, B.A.S.; Zainal, A.; Ghaleb, F.A.; Rassam, M.A. Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions. Appl. Sci. 2022, 12, 172. https://doi.org/10.3390/ app12010172.

[30] Craig Beaman , Ashley Barkworth, Toluwalope David Akande, Saqib Hakak, Muhammad Khurram Khan," Ransomware: Recent advances, analysis, challenges and future research directions", 24 September 2021.

[31] Faizan Ullah, Qaisar Javaid, Abdu Salam, Masood Ahmad, Nadeem Sarwar, Dilawar Shah, and Muhammad Abrar," Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls", 1 August 2020.

[32] Samuel Egunjobi, Simon Parkinson, and Andrew Crampton," Classifying Ransomware Using Machine Learning Algorithms", Department of Computer Science, School of Computing and Engineering, University of Hudders eld, Queensgate, Hudderseld HD1 3DH, UK

[33] Sagar Pande, Aditya Khamparia, Deepak Gupta, and Dang N. H. Thanh," DDOS Detection Using Machine Learning Technique", October 2020.

[34] Arshi M, Nasreen MD, and Karanam Madhavi," A Survey of DDOS Attacks Using Machine Learning Techniques", E3S Web of Conferences 184, 01052 (2020).

[35] Jiangtao Pei, Yunli Chen, Wei Ji, "A DDoS Attack Detection Method Based on Machine Learning", IOP Conf. Series: Journal of Physics: Conf. Series 1237 (2019) 032040.

[36] Swathi Sambangi * and Lakshmeeswari Gondi," A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression", 25 December 2020.

[37] Tuğba Aytaç, Muhammed Ali Aydın, Abdül Halim Zaim," Detection DDOS Attacks Using Machine Learning Methods", Electrica, 2020; 20(2): 159-167.