Final Review Document for Information Security Management Project

ASSIMILATION OF HONEYPOTS WITH MACHINE LEARNING TECHNIQUES FOR BETTER SECURITY

Name: Sri Chander Akunuri

Regd.No: 20BCE0523

Mobile No.: +91-8374253581

Mail Id: akunurisri.chander2020@vitstudent.ac.in

Name: Sree Harshavardhan Karanam

Regd.No: 20BDS0243

Mobile No.: +91- 7013950529

Mail Id: sree.harshavardhan2020@vitstudent.ac.in

Guide Name: Dr. Vimala Devi K

Designation: Associate Professor Sr.

Mobile No.: +91- 9842461744

Mail ID: vimaladevi.k@vit.ac.in

B. Tech

in

Computer Science and Engineering

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

ASSIMILATION OF HONEYPOTS WITH MACHINE LEARNING TECHNIQUES

Dr Vimala Devi K[1,*], Srichander Akunuri[1], Sree Harshavardhan Karanam[1]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore

*Project Guide

ABSTRACT: ***Information security is a major branch. It looks into how to protect a user's information in the most proper way so that attacks don't misuse it. It has various implementation to detect, trap or stop an attack from occurring. Honeypot is one such a implementation that tricks the attacker that they are attacking a "secure" port but that would be a trap laid down by the implementers to understand the technique of attacker and develop the information security protocols and methodologies to cater such a attack too in other words it could be said to be an information system resource whose value lies in unauthorized or illicit use of that resource. There have been various methodologies of implementing Honeypots like low interaction honeypots, high interaction honey pot, Dynamic Honey Pots in different cases and situation. In this project we will try to take the best from all the different scenario's and produce the best possible honeypot alongwith an audit assessment of its working.***

I. INTRODUCTION

Computers and other communication devices are of magnificent capability but also contain confidential information. Information that could be used in malicious ways and having devastating consequences if made open or accessible to the wrong people, and there are no less people out there who would love to access a vulnerability and do something with that information. Now, there are three methods traditionally that reduce the impact of such attacks when occurred- a. Detecting the attack, b. Stopping the attack i.e., not letting the attacker enter the secure zone or c. Preventing the attack but each of these methods would directly compromise the system data. Thus, as another layer of protection, a new layer/implementation was introduced. It is called the Honeypot. A honeypot is a trap to misguide the attacker from attacking the system and diverting them to a fake site so that the system owners can strengthen their core security system for such attacks. Thus, a honeypot is an efficient measure but, for this measure to work properly, we need to use the one that caters the needs of the system the most. Here, we try to design and build a honeypot system.

II. LITERATURE SURVEY

1. ***"Honeypots: Approach and Implementation" by Mayank Jain and Kumar Sridhar published on 12th December, 2014 in International Journal of Science and Research***

   In this paper the authors try to address the lack of information for starters on honeypot through the paper by discussing on the significance of a honeypot and various types of classifications in it. They discuss the location of implementation using various analysis and display methodologies. The only issue with the paper is that they don't explicitly discuss the design and compare various honeypots in a design sense.

2. ***"The Research and Design of Honeypot System Applied in the LAN Security" by Li Li, Hua Sun and Zhen yu Zhang published in 2012 in IEEE.***

   Through this paper, the authors try to address the lack of using the firewall to its full capacity and other detecting systems to secure a LAN to its highest possible extent. The authors for this propose to use a virtual Honeypot as well as a physical honeypot to efficiently lure the attackers and use firewalls and intrusion detection techniques to divert the intruder into the honeypot thus, thereby increasing the dataset to improve the security of the system. The problem with this paper is that, it doesn't represent a honeypot that by itself analyses and traps the intruder, rather a set of methods are discussed that trigger to push the intruder into the honeypot**.**

3. ***"The Dynamic Honeypot Design and Implementation Based on Honeyd" by Xuewu Liu, Lingyi Peng and Chaoliang Li published in 2012 in Springer.***

   In this paper the authors decided to address the issue mentioned earlier for the previous paper. Now, the authors of this paper try to present their ideas of the Isolation layer by integrating data science to learn the internet environment and the working of user. If a attacker is attacking to recognize the machine causing the attack and then diverting the attackers to the physical honeypot where they can be trapped. This paper has few issues though, it uses only Honeyd especially, which is a low interaction honeypot. It thus can't be used in the situations of high interaction effectively. It uses physical honeypot but doesn't mention explicitly on how to properly use it.

4. ***"A Dynamic Honeypot Design for Intrusion Detection" by Iyad Kuwatly, Malek Sraj, Zaid Al Masri and Hassan Artail published in 2005 in IEEE.***

   The paper discusses the problem that is same as the previous Springer one. It just discusses in extra on how to use create the physical honeypot and its rules.

5. ***"Heat-seeking honeypots: design and experience" byJohn P. John, Fang Yu, Yinglian Xe. Aravind Krishnamurthy and Martin Abadi published in March, 2011 in the University of Washington research and Microsoft research publications.***

   The paper implements a honeypot that puts the trap sites all over the net randomly to trap the attackers and reviews on the techniques and other statistics from the honeypot thus obtained.

III. HONEYPOTS

In computer terminology, a honeypot is a trap set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honeypot consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers. This is similar to the police baiting a criminal and then conducting undercover surveillance. Its primary purpose is not to be an ambush for the black hat community to catch them in action and to press charges against them. The focus lies on a silent collection of as much information as possible about their attack patterns, used programs, and the black hat community itself. All this information is used to learn more about the black hat proceedings and motives, as well as their technical knowledge and abilities. This is just a primary purpose of a honeypot. There are a lot other possibilities for a honeypot- divert hackers from productive systems or seize a hacker while conducting an attack are just two possible examples. Honeypots can be classified based on their deployment and based on their level of involvement.

A. CLASSIFICATION OF HONEYPOTS

Honeypots are classified based on various components like deployment.

Based on deployment

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots do. Production honeypots tend to mirror the production network of the company (or specific services), inviting attackers to interact with them in order to expose current vulnerabilities of the network. Uncovering these vulnerabilities and alerting administrators of attacks can provide early warning of attacks and help reduce the risk of intrusion. The data provided by the honeypot can be used to build better defences and counter measures against future threats.

Research honeypots are run to gather information about the motives and tactics of the Blackhat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats that organizations face and to learn how to better protect against those threats. Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations. Very little is contributed by a research honeypot to the direct security of an organization, although the lessons learned from one can be applied to improve attack prevention, detection, or response. They are typically used by organizations such as universities, governments, the military or large corporations interested in learning more about threats research.

Based on Level of Interaction

Low Interaction Honeypots don’t provide Operating system access to the intruder .It provides only services such as ftp ,http ,ssh etc. these low interaction honeypots play the role of passive IDS where the network traffic is not modified. The well-known example of low interaction honeypot is Honeyd. Honeyd is a daemon and it is used to simulate large network on a single host. It provides a framework to create several virtual hosts using unused IP addresses of the network with help of ARP daemon. For instance, several virtual number of operating systems, server, switches, routers, can be configured on a single host. Furthermore, emulated services include FTP service listening on port 21, login to FTP server etc. Other low interaction honeypot is specter and kFsensor. Specter can monitor total of 14 Tcp ports. Out of these fourteen ports seven ports are called traps and seven are called services. Traps act as a listener of ports i.e., when attacker makes connection with these ports the attempt is terminated and then logged. Services are more advanced wherever there is interaction between attacker and emulating services.

High Interaction Honeypots These are the most sophisticated honeypots .These are difficult to design and implementation. These honeypots are very time consuming to develop and have highest risks involved with this as they involve actual OS with them .In high Interaction Honeypots nothing is simulated or restricted. A few examples of High interaction honeypots are Sebek and Argos. As these honeypots involves real operating system the level of risk is increased by many extents, but to capture large amount of information by allowing an attacker to interact with the real operating system, it is a kind of trade off. This helps in capturing and logging of attacker’s behaviour that can be analysed in later stage.

B. DEPLOYMENT OF HONEYPOT

First, we create a bunch of honeypots or honeynets which lures the attacker to attack that network. The ports must be left open accordingly so as to make attackers believe it not to be a trap. A genuine user can be distinguished from an attacker depending whether he access the data or the honeypot. Firewalls are employed in the whole network to increase its security. Web server, mail server, client etc. are forwarded to the legitimate destination and honeypot fulfil the task of luring the attacker. Standard mechanisms are used for protection of web and mail servers. Services such as web, mail, ftp services and DNS that should be accessible from the out- side are situated in a demilitarized zone (DMZ).

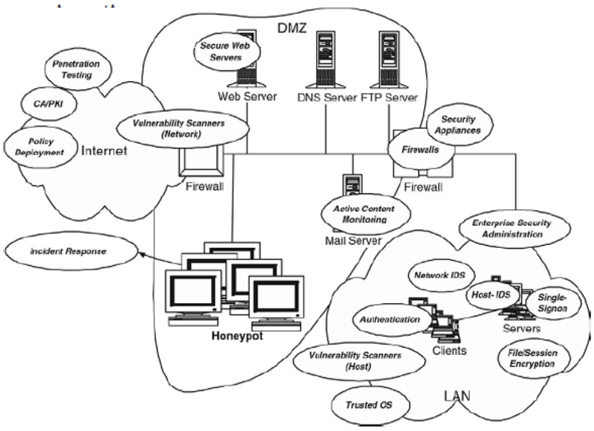


Figure 1: The implementation of Honeypot

In the above figure we create a Data Management Zone (DMZ) where we put our Data Servers, DNS servers and FTP Servers and we don’t want the attacker to attack our some of the most important system files.

C. USES

Honeypots are used to know the system users or attackers better. A typical web server may get millions of hits every day. So, it gets quite difficult to identify the attackers from the users. So, we put honeypots in our same network system. Honeypots are just used to lure the attackers. If we get hits on our honeypots, we can distinguish the attacker from the users. As honeypots have no legitimate uses, we trace back the attacker and improve our network security.

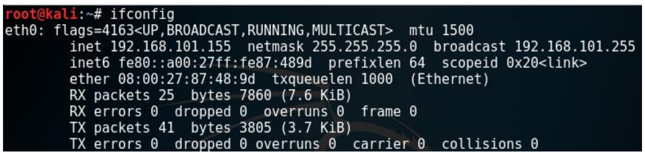
IV. PROPOSED METHODOLOGY

I've noticed that many of the papers don't have a proper assessment even with good ideas and. ML technologies are seldom used. I propose the use of ML techniques for a better security. The proposed methodology has the following major steps for implementation.

1. *Use Nmap or Snort to probe and fingerprint the user/ host details.*
2. *Use Pentbox in Kali Linux to emulate a honeypot. It also provides intrusion details as well as generates a honeypot at the given port.*
3. *Try to implement ML techniques to divert traffic from main computer if detected to the free port.*
4. *All of this data will be stored in a database Alongwith other log on files.*
5. *Assess the working of the honeypot by simulating attacks.*

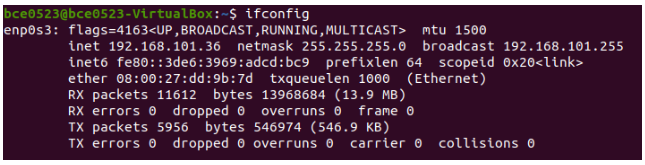
V. IMPLEMENTATION

IP ADDRESS IDENTIFICATION



```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.101.155  netmask 255.255.255.0  broadcast 192.168.101.255
        inet6 fe80::a00:27ff:fe87:489d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:87:48:9d  txqueuelen 1000  (Ethernet)
        RX packets 25  bytes 7860 (7.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41  bytes 3805 (3.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

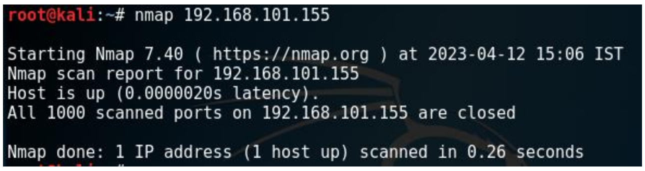
Figure 2a: IP configuration of the UNIX machine being attacked



```
bce0523@bce0523-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.101.36  netmask 255.255.255.0  broadcast 192.168.101.255
        inet6 fe80::3de6:3969:adcd:bc9  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:dd:9b:7d  txqueuelen 1000  (Ethernet)
        RX packets 11612  bytes 13968684 (13.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5956  bytes 546974 (546.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

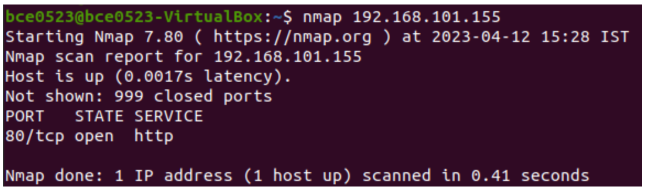
Figure 2b: IP configuration of the UNIX machine attacking

NMAP OF THE COMPUTER BEING ATTACKED



```
root@kali:~# nmap 192.168.101.155

Starting Nmap 7.40 ( https://nmap.org ) at 2023-04-12 15:06 IST
Nmap scan report for 192.168.101.155
Host is up (0.0000020s latency).
All 1000 scanned ports on 192.168.101.155 are closed

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
```

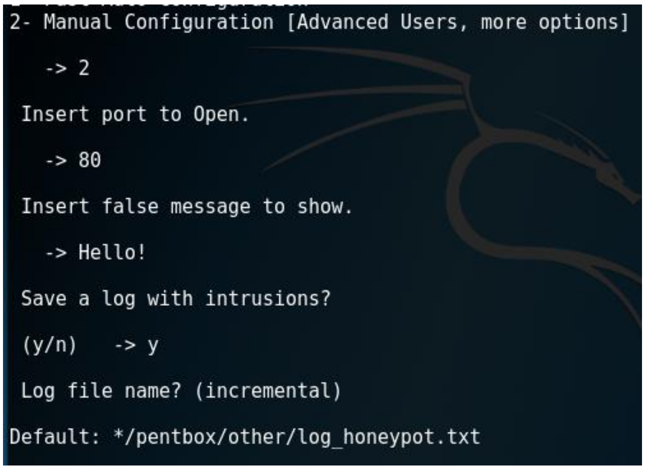
Figure 3a: NMAP before deployment of honeypot



```
bce0523@bce0523-VirtualBox:~$ nmap 192.168.101.155
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-12 15:28 IST
Nmap scan report for 192.168.101.155
Host is up (0.0017s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
```

Figure 3b: NMAP after the deployment of Honeypot on Port 80

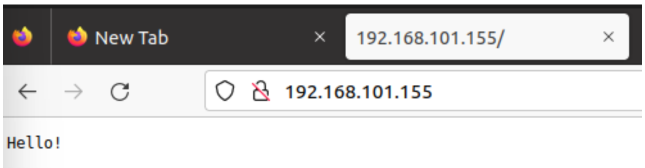
DEPLOYMENT OF HONEYPOT



```
2- Manual Configuration [Advanced Users, more options]

   -> 2

 Insert port to Open.

   -> 80

 Insert false message to show.

   -> Hello!

 Save a log with intrusions?

 (y/n)   -> y

 Log file name? (incremental)

Default: */pentbox/other/log_honeypot.txt
```

Figure 4: Opening the port 80 and inserting the message 'Hello!' to be displayed when accessed.

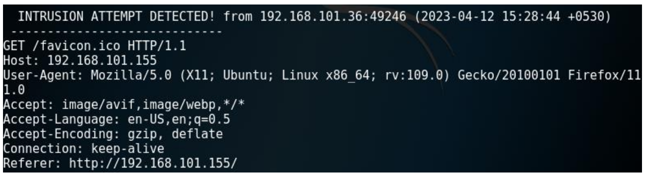
ATTACKER ACCESSING THE PORT



```
Hello!
```

Figure 5: Port 80 is default HTTP port. Thus, the above is obtained when accessed.

LOG FILE GENERATED FOR THE ABOVE ACCESS

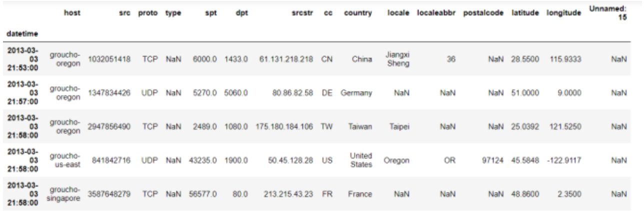


```
 INTRUSION ATTEMPT DETECTED! from 192.168.101.36:49246 (2023-04-12 15:28:44 +0530)
 -----------------------------
GET /favicon.ico HTTP/1.1
Host: 192.168.101.155
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/11
1.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.101.155/
```

Figure 6: The above log generated gives all the basic details obtainable from the intrusion.

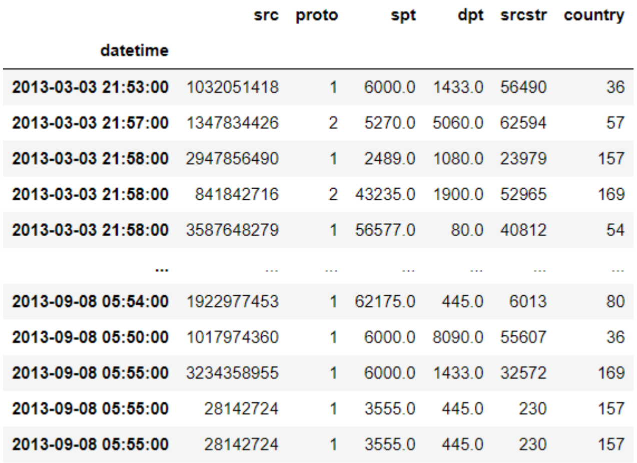
MACHINE LEARNING TECHNIQUES

For the ML implementation, due to the lack of we have a data of the intrusions of a cloud honeypot by AWS which we were not able to use to predict in our honeypot situation.



	host	src	proto	type	spt	dpt	srcstr	cc	country	locale	localeabbr	postalcode	latitude	longitude	Unnamed: 15
datetime															
2013-03-03 21:53:00	groucho-oregon	1032051418	TCP	NaN	6000.0	1433.0	61.131.218.218	CN	China	Jiangxi Sheng	36	NaN	28.5500	115.9333	NaN
2013-03-03 21:57:00	groucho-oregon	1347834426	UDP	NaN	5270.0	5060.0	80.86.82.58	DE	Germany	NaN	NaN	NaN	51.0000	9.0000	NaN
2013-03-03 21:58:00	groucho-oregon	2947856490	TCP	NaN	2489.0	1080.0	175.180.184.106	TW	Taiwan	Taipei	NaN	NaN	25.0392	121.5250	NaN
2013-03-03 21:58:00	groucho-us-east	841842716	UDP	NaN	43235.0	1900.0	50.45.128.28	US	United States	Oregon	OR	97124	45.5848	-122.9117	NaN
2013-03-03 21:58:00	groucho-singapore	3587648279	TCP	NaN	56577.0	80.0	213.215.43.23	FR	France	NaN	NaN	NaN	48.8600	2.3500	NaN

Figure 7: AWS Data



	src	proto	spt	dpt	srcstr	country
datetime						
2013-03-03 21:53:00	1032051418	1	6000.0	1433.0	56490	36
2013-03-03 21:57:00	1347834426	2	5270.0	5060.0	62594	57
2013-03-03 21:58:00	2947856490	1	2489.0	1080.0	23979	157
2013-03-03 21:58:00	841842716	2	43235.0	1900.0	52965	169
2013-03-03 21:58:00	3587648279	1	56577.0	80.0	40812	54
...	...	...	...	...	...	...
2013-09-08 05:54:00	1922977453	1	62175.0	445.0	6013	80
2013-09-08 05:50:00	1017974360	1	6000.0	8090.0	55607	36
2013-09-08 05:55:00	3234358955	1	6000.0	1433.0	32572	169
2013-09-08 05:55:00	28142724	1	3555.0	445.0	230	157
2013-09-08 05:55:00	28142724	1	3555.0	445.0	230	157

Figure 8: Post Processed Data

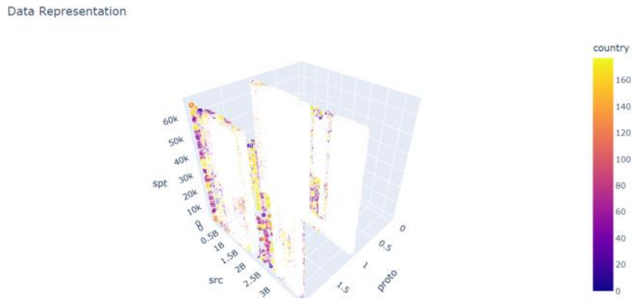
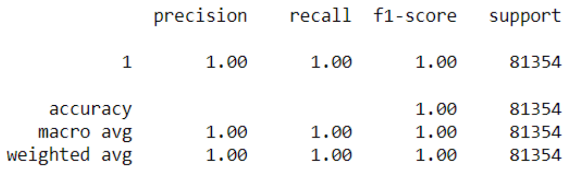


Figure 9: 5-D Data Representation for the data in Figure 8



```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00     81354

    accuracy                           1.00     81354
   macro avg       1.00      1.00      1.00     81354
weighted avg       1.00      1.00      1.00     81354
```

Figure 10: Naïve Bayes classifier result showing the accuracy as 100% which is impossible and wrong in the given scenario. Thus, we opt for clustering techniques

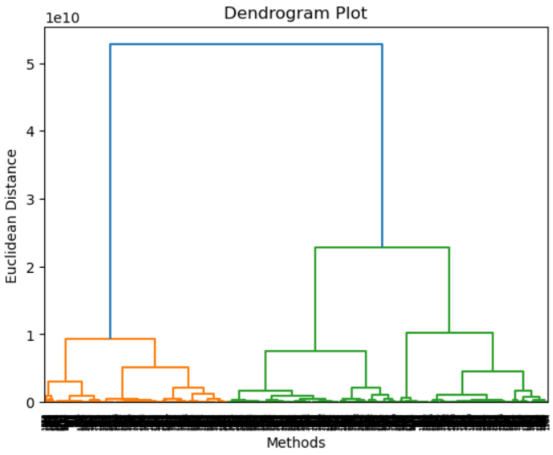
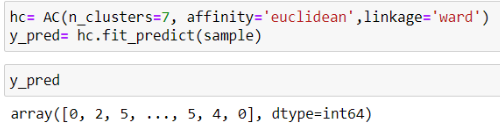


Figure 11: Dendrogram plot for Agglomerative Hierarchical clustering



```
hc= AC(n_clusters=7, affinity='euclidean',linkage='ward')
y_pred= hc.fit_predict(sample)
```

```
y_pred
```

```
array([0, 2, 5, ..., 5, 4, 0], dtype=int64)
```

Figure 12: The output clustering of Agglomerative.

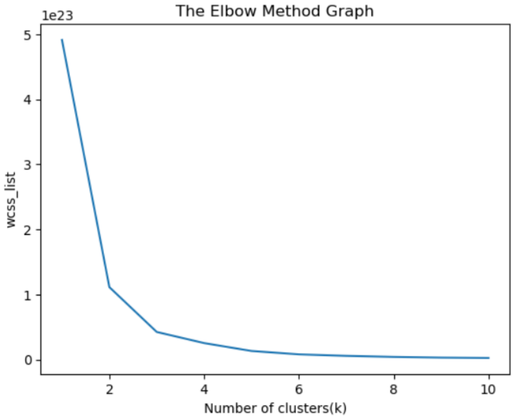
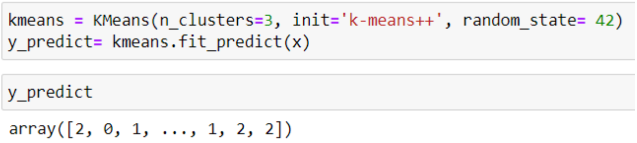


Figure 13: We later moved to K-Means as Hierarchical clustering was too resource consuming for the whole data. From the above elbow graph, it is evident that the optimum number of clusters will be 3.



```
kmeans = KMeans(n_clusters=3, init='k-means++', random_state= 42)
y_predict= kmeans.fit_predict(x)
```

```
y_predict
```

```
array([2, 0, 1, ..., 1, 2, 2])
```

Figure 14: The K-Means clustering output

VI. AUDIT REPORT

PURPOSE AND SCOPE

The purpose of this Audit is to create a proper assessment on the Pentbox honeypot and thus we are limiting our scope to this only.

OBJECTIVES

1. Verify the working of honeypot in various situations.
2. Check if proper logging generated data is being done.
3. Detecting honeypot vulnerabilities

CONSTRAINTS

1. Using a personal network
2. Use of a Virtual Machine
3. Time Constraint

COMPONENTS AND DEFINITIONS

1. Personal Network
2. UBUNTU Virtual Machine
3. KALI Virtual Machine
4. Network security testing : Network testing is a broad means of testing security controls across a network to identify and demonstrate vulnerabilities and determine risks. While the testing medium can change (wireless, ethernet, hardware/IoT, phishing emails, physical access, Dropbox placement), the end result is usually network access to protected data or systems.
5. Vulnerability - a vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries (i.e., perform unauthorized actions) within a computer system. To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerabilities are also known as the attack surface.

AUDITING PHASES

1. No specific phase methodology used.
2. Obtained data as we proceed in the construction of the honeypot directly.

AUDITING TASKS:

a. Network Scanning
b. Vulnerability Scanning
c. Password Cracking [Not applicable]
d. Log Review
e. File Integrity Check
f. Virus Detectors [Not Applicable]
g. Wireless N/W Testing
h. Authentication
i. Session Management
j. Access Control
k. Communication Security [Not Applicable]
l. Business Logic [Not Applicable]
m. Backup Storage [Not Applicable]
n. Disaster Recovery Plan Implementation [Not Applicable]
o. Configuration Documentation

AUDIT METHODOLOGIES

1. Observation of working
2. Risk assessment

[Other forms not applicable]

AUDIT REPORT

Refer Table 1

VII. CONCLUSION

Honeypots are devices with enormous capabilities in terms for security testing. They can be integrated with the present day ML technologies through proper means for providing high security with capable amount of integrity of the files and the product involved without compromising the data security of the parties involved in a well designed way and its not impossible.

AUDIT REPORT:

Task Performed	***Check***	***Findings***	***Suggestion***
Network Scanning [Also involves Wireless N/w Testing]	Nmap of Honeypot Host Pre and Post implementation	Ports opened were as required and no issue post deployment	None
	Nmap check of the attacker	Ports opened were as required	None
Vulnerability Scanning	The vulnerability of the VM machine before and after deployment of Honeypot and in the intermediate	The VM's port wise vulnerability is near null and only the port that was expected to be operating was visible. The only vulnerability of honeypot was that it was bad in implementing FTP on its own which would for sure alert the attacker in a real situation. No issues observed in the intermediate stage otherwise.	FTP etc should be given an option to add a path to a web file to give the attacker a mirage of a real FTP implementation.
Log Review	If the log was being recorded correctly	The log was corresponding correctly. No anomalies found but not tabular in nature but more of a text	Log could be saved in a tabular approach to make it easier to access in a urgent situation.
File Integrity Check	If the Honeypot files are properly arranged in the folder	All are properly named according to their usage and no ambiguity observed.	None
Authentication	If any Anomaly in authentication is observed	We didn't observe any authentication anomaly. The honeypot allowed everyone and every device as it was intended to.	None
Session Management	If the session is properly managed or not by the Honeypot	To end a session, we need to close the whole terminal and it closes the honeypot on that specific port. This causes possibility of the honeypot deployment on various ports difficult and multiple terminal instances need to be instantiated to make it possible which makes management of honeypots difficult and thus could cause a issue during a major attack observation	Try to integrate a loop that would take at the start itself, the list of ports on which the deployment could be done. So that many other honeypots could be deployed on all the ports simultaneously on the same terminal.
Access Control	If any invalid access is being provided	No such suspicious activity found	None
Configuration Documentation	If configuration documentation is well and proper	The configuration documentation is well and simple to understand and follow.	None

Table 1: Audit Report