International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 07 | July 2019 www.irjet.net p-ISSN: 2395-0072

A CLOUD BASED HONEYNET SYSTEM FOR ATTACK DETECTION USING MACHINE LEARNING TECHNIQUES

Mareena Marydas[1], Varshapriya J N[2]

[1]*Student, Dept. of Computer Engineering, VJTI, Mumbai, India*
[2]*Professor, Dept. of Computer Engineering, VJTI, Mumbai, India*

Abstract - *The Cloud network nowadays, are compromised and exploited by hackers. Large enterprise networks, such as the network for a major university or any organization, behave as enticing targets to be exploited by intruders. The requirements for user freedom, prohibits the system administrator to impose any kind of restrictions on the user. And thus, it is important to secure our network, for discovering unwanted activities, and build a trap system like the honeypots. A honeypot is a well monitored network deception tool designed to serve several purposes: it can distract nemesis from various interconnections in a network, layout an early signal about new attacks and exploitation trends and allow thorough and clear picture of the nemesis during and after the exploitation of the honeypot. A Honeynet is a collection of one or more Honeypots. We propose a Honeynet system that emphasizes on detecting any attack, or suspicious activity on protocols like SSH, FTP etc. in our cloud network. These techniques can be heuristically used to determine the difference between Malicious and benign traffic. Also, this is a comparative study as to which one of these Classification algorithms would give us a low false positive rate*.

Key Words: **Cloud Computing, Honeynet, Classification Algorithms, Openstack, Cloud instance**

1. INTRODUCTION

Cloud computing has been defined by NIST as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be quickly provisioned and divulged with lesser effort and interaction of Service providers[1]. Although reducing cost is a primary goal of cloud providers, reducing the security shouldn't be. Monitoring and dealing with security issues remain the priority of any organization, just as other important issues, such as performance, availability, and recovery[2]. Most of the incidents that occur on the cloud are not to exploit any organization or its services. Instead, their objective is the vulnerable victim.

At present it is not essential to know the way a system functions, in order to be able to attack it therefore, present-day companies require solutions that will not only understand the tactics of the attacker but also supervise the users who have gained access to the cloud system. [4]

Lance Spitzner, the founder of The Honeynet Project organization, defines a honeypot as: "Honeypot could be security resource whose worth lies in being probed, attacked or compromised" [5]. A Honeynet is a collection of honeypots which is a decoy like system that is intended to be compromised, so as to study the intensions of an attacker and to detect whether the system has been compromised or attacked.

A Honeynet is a network, placed behind a firewall that that encapsulates the ingress and the egress data. This data is contained, examined, and controlled. Any type of system can be placed within the Honeynet, to mimic the behavior of the real systems we intend to protect. Other security tools such as IDS and firewalls are passive in nature, as their task is to detect and prevent attacks. Honeypots that are deployed in our cloud give a way to the attacker to actively intrude the system.

As in [4], one of the deployment issues is the number of honeypots to be configured. Thus we propose a system that consists of collection of various honeypots namely Dionaea[10], Cowrie[11] and Honeytrap[12] which create a honeynet. Such a honeynet will be used to capture traffic from ports of different machines on the internet, which are trying to gain access into our cloud system. The main issue with the logs generated by the honeypot is the huge amount of data that has to be analyzed and studied by a human expert[13]. Hence the logs fetched from our honeynet will then be effectively used as training set inputs to three classification algorithms viz. SVM, Random Forest and Naïve Bayes. The output would then help us determine a suitable machine learning algorithm to a particular honeypot according to accuracy. The proposed model is illustrated in Fig.1.

We have created the private cloud environment by using Openstack, launched two virtual machine instances on Openstack, one being the attacker and one having the honeypots configured to form a honeynet.

This paper is organized as follows: Section 2 describes the review of Literature, Section 3 consists of information about our Proposed system which further has subsections describing about the Cloud environment, simulation and traffic capture. The data logs gathered are analyzed and studied in a tabular form in section 4. Lastly, we conclude this research with future scope in Section 5.

2. REVIEW OF LITERATURE

The first line of defense in the security of data, during early times was to keep away the intruders exclusively. Since then various tools and technologies have been developed diligently to avoid attacks. And hence in 1992, Marcus Ranum developed the Firewall.[6]

System administrators protect the security of their networks in numerous ways. The use of Firewall to filter the traffic from the Internet, before it enters their network is one such method[6]. Firewalls help in protecting the organizations and stop attackers from exploiting the system[7]. Intrusion Detection Systems (IDS) are another example of such tools allowing administrators to detect and identify attacks or malicious events by an intruder. But these tools at times, lack the capability of identifying new attacks or collection of data regarding the attacker or the attacker's motive, which is very essential these days. The problem with firewalls or IDS is that it involves high amount of false negative or false positive alerts. Honeypot when deployed behind the firewall, serves as an in-depth defense system, which can not only log the attackers who get through the firewall but also detect any kind of perils from the insiders. [8]

As in [3], Honeypots could be categorized as Low Interaction, Medium Interaction and High Interaction:

i. Low interaction honeypots(Production honeypots): They have limited interaction with the attacker and are simple to deploy. They work by emulation of services and systems. A production honeypot is used to aid an organization in securing its internal infrastructure. They generally do not give as much information as research honeypots.

Examples of low interaction honeypot include KFSensor, Honeyd, and Specter.

ii. Medium interaction honeypots allows some kind of interaction with the attacker (i.e. bash shell). The only difference is that attackers feel that they have gained access into the system. But in reality they are just interacting with the emulated shell.

Examples include: Kippo, Cowrie, Dionaea, hornet.

iii. High Interaction honeypots(Research honeypots): The primary mission of research honeypot is to research or explore the threats organization may face, such as who and how the attackers are, how they are organized, what kind of tools they use to attack other systems, and where did they obtain the tools from.

Examples of high interaction honeypot include Symantec Decoy Server, Honeytrap and Honeynets.

As the attacks and the attackers grow wiser and powerful each day with new exploits, it is important to secure our network loop-holes. The thesis mentioned in [13] gives a different dimension to implementation of different kinds of honeypots. It focuses on an SMB Honeypot which is used for collection of malwares like virus or worms. SMB is a widely used protocol to propagate these attacks. Another type of honeypot is the Honeypot-db. This honeypot logs all is able to log all the traffic to and from the MySQL server with the victims IP. The Honeypot can be configured even as a Web Server[14]. The WS Honeypot is a high interaction honeypot. It creates an interface which provides proper web services, so that an intruder feels like an interaction with the real web server. To automate the process of collecting and analyzing logs from the WS Honeypot, they've focused on developing machine learning techniques like SVM(Support Vector Machine).

Machine learning or data mining techniques like Classification, Clustering etc. have been extensively used these days due to the high amount of traffic logs generated by the honeypot. It becomes extremely important to distinguish between normal traffic and malicious traffic. Thus various data mining techniques have been proposed to extract useful and critical information from large databases.[14] The classification relies on a learning dataset built progressively that arranges a data item into predefined groups. Clustering places data items into related groups that are unknown and it is up to the clustering algorithm to find out most adaptable classes. Also another method of data mining so used are the Association Rules.

Another method of deploying a honeypot is proposed in [15] which is to deploy a Honeypot Router i.e. a honeypot playing the role of a router. Such a honeypot is used to analyze and study about attack patterns against routing protocols like OSPF, RIP and BGP. And the data mining algorithm used to analyze the logs was the DBScan Clustering algorithm. DBScan was chosen over K-mean and Cobweb clustering algorithms, because it gave lesser false positives after testing.

Botnets are a set of compromised computers(malware) which can be infected to attack other systems in the network. To understand these attacks by the data collected, a clustering structured visualization technique with outlier detection is proposed in [16]. The algorithm used is the K-Nearest Neighbour (KNN), with a local outlier definition, Local Outlier Factor (LOF), which has together been called as KNOF, which is used for outlier detection to distinguish between malicious and benign traffic.

In [17] classification techniques like SVM, Decision Tree Method and Random forest are used to classify malware according to their static and dynamic features. These features are extracted and compared in terms of accuracy and time. Among these, Random Forest Algorithm yields maximum accuracy. It is very important these days to reduce the rate of false positives and false negatives while

implementing any machine learning algorithm to improve accuracy and yield proper results. In [18] the paper is focused on complementary approaches to reduce the number of false positives in Intrusion Detection. It is reduced by alert post-processing. They have verified and tested it on both simulated and real Environments. And achieved a significant decrease in the number of false positives.

3. PROPOSED SYSTEM

3.1 Cloud Environment

We have created a private cloud environment by using Openstack to implement and test our proposed system. Openstack enlarged into large community with more than 9000 organizations and 500 companies [19]. Openstack can be defined as open source software platform which behaves

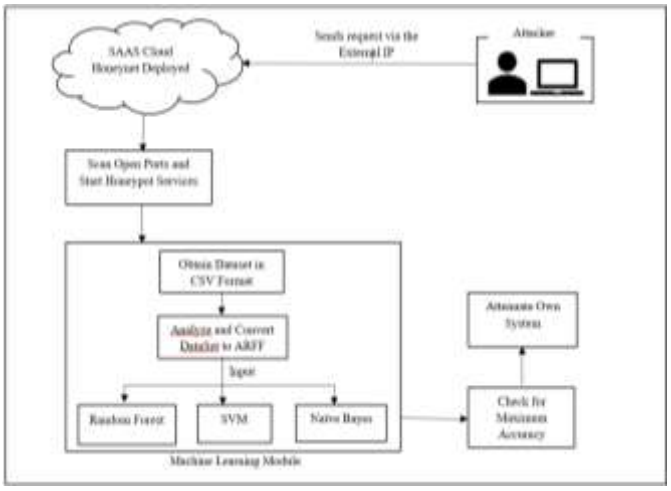

Fig -1: Architecture of our Honeynet

Configured as Controller(Keystone) Networking(Neutron) and Compute(Cinder) nodes respectively[21].

As proposed all the three honeypots will be configured, for which we will need an Ubuntu OS. So we launch an Ubuntu Instance after the installation of Openstack.

3.2 System Description

Any machine or system, trying to connect or communicate with the cloud instance, sends a normal HTTP request initially. Numerous victim machines can be splayed out in the Honeynet and can execute different operating systems. When requests are sent from one machine to another, streams of packets are sent and received by the external IP of the cloud. The data is gathered and sent for future pre-processing into arff. format. This arff. Data format is then given as input to three machine learning algorithms to furnish maximum accuracy.

as a cloud and provides computing facilities specially infrastructure service, where customers can deploy their own instance (virtual machine) on cloud [20].

Each of the services in Openstack provides an interface. It depends on our needs whether we require some, or all services. It is an open source software for creating private and public clouds, built and disseminated by a large and democratic community of developers, in collaboration with users.

Openstack is an SDN(Software Defined Networking Architecture) made up of different components such as Controller, Compute and Network etc. These components can be installed separately in different PC's or all within a single machine. We have configured the multinode Openstack environment with three PCs

Our paper focuses on a comparative study of honeypots which when deployed with machine learning algorithms give us a good accuracy rate. The three honeypots that form a honeynet namely are:

i. Cowrie honeypot

Cowrie is an SSH honeypot which attempts to masquerade an SSH Server specifically a server with weak login password credentials. The logs collected depend on SSH logins by the attacker.

ii. Dionaea Honeypot

It emulates a gullible Windows system with provided services often targeted by intruders such as FTP HTTP, SMB, etc. Dionaea forwards real time notifications via XMPP and gathers its logs in SQLite database.

iii. Honeytrap Honeypot

Honeytrap is a framework used for managing sensors, low interaction, medium interaction and high interaction honeypots together. Honeytrap consists of services, channels, directors and listeners. This honeypot executes a dynamic server concept. It scans the network for incoming traffic and starts suitable listeners, which handle many connections.

3.3 Honeynet Simulation

Once the Honeypots are deployed on the cloud instance, we start our Instance using the External IP of the cloud, in the Browser, with the appropriate port number. Here we create two instances, one the attacker instance and second one the victim instance on which our honeypots are deployed. And generate an SSH key for exchanging information.

We simulate both the attacker and the victim machine in our cloud instance and present the results accordingly. The attacker instance will be running the Metasploit Framework through which we will be experimenting with various

exploits on our victim cloud instance where the honeypots are deployed.

3.4 Traffic Capture

As proposed, the attacker instance gets an IP **34.68.7.193** and the victim/honeypot instance External IP is **35.222.214.147.**

So firstly we do an Nmap scan on the victim machine to find out open ports(Fig. 2)

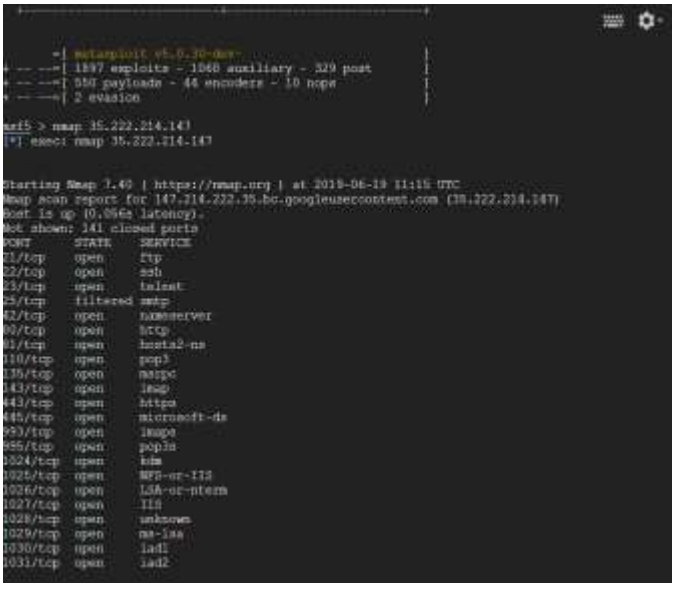

Fig -2: Screenshot of NMAP scan on Victim

The open ports found are SSH, Telnet, FTP etc. We demonstrate an ssh login attack(Fig. 3), which will be logged by the Cowrie honeypot(Fig. 4).

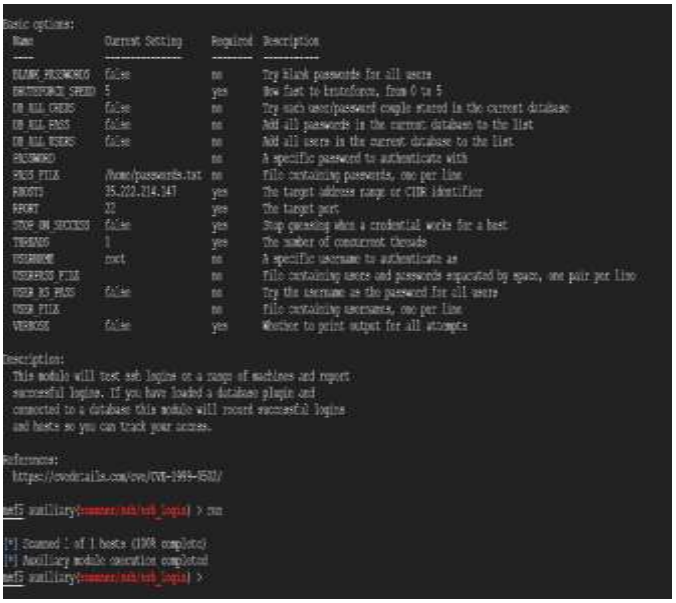

Fig -3: Screenshot of SSH Login Attack on Victim

Fig -4: Screenshot of Cowrie Honeypot UI

Similarly other attacks on protocols like FTP, Http, Smbd etc are performed and logged by Dionaea and Honeytrap honeypots. The logs so formed, are captured to form datasets. These datasets are imported in CSV format.

4. ANALYSIS AND RESULTS

In [14], it is stated that, to fetch good results from the honeypots, the data collected, should first undergo a training phase. In this phase, we let the machine learn according to normal http requests that are not attacks. The training phase takes place in a safe environment. The activities will be classified according to their "message" field in the dataset. Once the machine is trained, according to the selected attributes in WEKA, then the Machine learning algorithms are run, so as to obtain max accuracy.

Machine Learning Module:

i. Naïve Bayes Classification Algorithm

Naive Bayes is a classification technique which assumes that a particular feature in a dataset are independent of the other features. For example, considering a fruit, particularly an orange. Features like its color (orange), shape (round) and diameter (3 inches) are independent from other samples in the dataset, which contributes to a greater probability that this fruit is an orange and hence it's called Naïve.

Naïve Bayes performs better in case of the huge size of data, since they execute parallel map-reduce implementation on it, also because they are simple to train. Thus, this makes it quicker than Random Forest and SVM. Even in our system, when trained, Naïve Bayes takes lesser time to train the model and test it. Although it gives lesser accuracy rate than the others.

Table -1: Comparison of Naïve Bayes Performances with respect to honeypots in Honeynet

	Accuracy	True Positive	False Positive	Time (Test Model)
Cowrie	97.80	0.97	0.109	0.03
Dionaea	97.84	0.97	0.28	0.01
Honeytrap	94.56	0.94	0.176	0.02

ii. SVM: Support Vector Machine

SVM is a supervised machine learning algorithm that is descriptive in nature as compared to Naïve Bayes which is a generative model. It is based on a function given by y = w.x+b, where weight (w) and bias (b) are estimated from the training data. SVM minimizes the generalization error rather than minimizing the training error [19].

As compared to performance, SVM performs better than Naïve Bayes since it forms a hyperplane that maximizes the margin of learning. In general, the SVM takes more time to train than Naïve Bayes, but the predictions are more accurate.

Table -2: Comparison of SVM Performances with respect to honeypots in Honeynet

	Accuracy	True Positive	False Positive	Time (Test Model)
Cowrie	98.75	0.98	0.108	0.66
Dionaea	98.05	0.99	0.274	0.21
Honeytrap	97.44	0.97	0.168	13.29

iii. Random Forest

Random Forest is a supervised learning algorithm which is a collection of Decision Trees. It is robust against overfitting and works good with numerical data. It gives importance to features, because it measures the impact each predictor has on the final results.

The great disadvantage of this algorithm is that, a high number of trees may make the calculation and training process slower and ineffective for real-time predictions. Even with our system, it takes a sufficiently longer time than Naïve Bayes and SVM. Its accuracy is greater than Naïve Bayes in comparison.

Table -3: Comparison of Random Forest Performances with respect to honeypots in Honeynet

	Accuracy	True Positive	False Positive	Time (Test Model)
Cowrie	98.87	0.98	0.1	51.72
Dionaea	98.59	0.98	0.12	0.29
Honeytrap	98.37	0.98	0.98	7.5

5. CONCLUSION

In this paper, we presented a Honeynet system consisting of three robust honeypots to detect attacks. To study the logged activities by this Honeynet, we experimented with three machine learning learning techniques. We have trained the models with Naïve Bayes, SVM and Random Forest, so that new incoming data from these honeynets can be classified as malicious with greater accuracy. The results so obtained have been detailed in the paper.

As future work, we focus on reducing false negatives by better better association with knowledge from the test environment. We also plan to add other functionalities as to automate the system more efficiently to reduce human efforts and increase precision.

REFERENCES

[1] P. Mell, T. Grance, The NIST Definition of Cloud Computing, Version 15, National Institute of Standards and Technology, October 7, 2009, http://csrc.nist.gov/groups/SNS/cloud-computing.

[2] Jansen, W. A. (2011). Cloud Hooks: Security and Privacy Issues in Cloud Computing. 2011 44th Hawaii International Conference on System Sciences.

[3] Deepak R Bharadwaj, Anamika Bhattacharya and Manivannan Chakkaravarthy "Cloud Threat Defense – a Threat Protection and Security Compliance Solution", *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2018.*

[4] Kuwatly, I., Sraj, M., Al Masri, Z., and Artail, H. (2004) 'A dynamic honeypot design for intrusion detection', In Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on (pp. 95-104). IEEE.

[5] Spitzner, L.Honeypot: Definitions and Values. May, 2002. http://www.spitzner.net.

[6] Velasco Silva, D. and, Rodríguez Rafael, G.D. 'A Review of the Current State of Honeynet Architectures and Tools', Int. J. Security and Networks, Vol. X, No. Y, pp.xx–xx

[7] Levine, J., La Bella, R., Owen, H., Contis, D., & Culver, B. (n.d.). "The use of Honeynets to detect exploited systems across large enterprise networks". IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003.

[8] Aliyev, Vusal. "Using honeypots to study skill level of attackers based on the exploited vulnerabilities in the network." Department of Computer Science and Engineering Division of Computer Security CHALMERS UNIVERSITY OF TECHNOLOGY Göteborg, Sweden, 2010.

[9] Chao-Hsi Yeh, & Chung-Huang Yang. (2008*).* "Design and implementation of honeypot systems based on open-source software". 2008 IEEE International Conference on Intelligence and Security Informatics.

[10] Dionaea website.:Honeypots CERT Exercise Handbook: Honeypots CERT Exercise Handbook

[11] Cowrie Honeypot: https://null-byte.wonderhowto.com/how to/use-cowrie-ssh-honeypot-catch-attackers-your-network 0181600/

[12]Honeytrap : http://honeytrap.carnivore.it/documentation/

[13] Nkwetta Jeffrey Asonganyi.(2018) "Honey-system: design, implementation and attack analysis." College of Technology,University of Buea.

[14] Abdallah Ghourabi, Tarek Abbes, Adel Bouhoula, "Design and Implementation of Web Service Honeypot," IEEE, SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks.

[15] Ghourabi, A., Abbes, T., & Bouhoula, A. (2010). "Data analyzer based on data mining for Honeypot Router," ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010.

[16] H. Jin, O. Vel, K. Zhang and N. Liu, "Knowledge Discovery from Honeypot Data for Monitoring Malicious Attacks," In Proceedings of the 21st Australasian Joint Conference on Artificial intelligence 2008, pp. 470-48

[17] Katerina Goseva-Popstojanova, et Al. "Characterization and classification of malicious Web traffic" in Computer and Network Security, Vol: 42, pp. 92-115, 2014.

[18] Tadeusz Pietraszeka, Axel Tanner, "Data Mining and Machine Learning—Towards Reducing False Positives in Intrusion Detection." ACM Journal, Information Security Tech. Report. Volume 10 Issue 3, January, 2005

[19] V. N. Vapnik, "The Nature of Statistical Learning Theory," SpringerVerlag New York, Inc., New York, NY, USA, 1995

[20] Openstack : https://vmokshagroup.com/tag/openstack/

[21] Openstack: https://en.wikipedia.org/wiki/OpenStack

[22] https://www.techsupportpk.com/2016/12/installing-openstack-on-multi-node-in-linux.html

[23] https://www.linuxtechi.com/launch-instance-from openstack-dashboard/

[24] https://www.geeksforgeeks.org/how-to-setup-firewall-in-linux

[25] Weka: https://www.cs.vassar.edu/~cs366/docs/weka-tutorial-full.pdf

[26] https://medium.com/machine-learning-101

[27] https://elitedatascience.com/machine-learning-algorithms