VEGUETA Vol 9 Issue 01,May 2021
ISSN NO: 1133-598X

Honey Pots and Machine Learning: Defending Against Distributed Denial of Service Attacks from Botnets on the Internet of Things

[1]*M. Sreenivasulu,* [2] *Nageswara Rao Sirisala,* [3]*N. Ramanjaneya Reddy [4] Dr V Venkata* [2,3] [1,4]
Professor, Associate Professor,

Department of CSE, K.S.R.M College of Engineering(A), Kadapa

Ramana

Abstract—
IoT botnet DDoS attacks have skyrocketed in frequency and severity in recent years, making IoT security an urgent concern for system administrators worldwide. Many different approaches to Internet of Things security have been proposed, but none have proven effective against the growing prevalence of Zero-Day Attacks. In this research, we provide a honey pot technique that takes use of ML for malware detection. The data collected from IoT honey pots is used to rapidly and effectively fine-tune a machine learning model. Dodos attacks, which have lately emerged as a big concern, may be prevented in part by using this technique on the IoT.

I. INTRODUCTION

Now, the Internet of Things (Iota)—a system of interconnected physical objects that may function without human oversight—may be the source of Dodos attacks [1]. More desktop PCs are being used because of the ease of Internet of Things devices. As a result, IoT-based bonnet attacks have increased dramatically [7]. When malware infects an IoT network, it creates what's called a "bonnet," or a group of bots (hijacked IoT gadgets) [2]. Hackers can't afford to ignore the industry any longer, as a recent study indicated that there are more than 6 billion IoT devices in use globally. Malware has been discovered on a massive scale throughout the years, with 2017 alone accounting for more than half of all detections [5]. A honeypot may be used to observe and study an attacker's method of launching an attack by capturing data about the attacking agent, such as malware for a DDoS assault [9]. It might pretend to be the main server and be hacked by imitating any vulnerability that is easily exploited by an attacker. Monitoring the communication between the attacker and itself might reveal sensitive information such as IP addresses, MAC addresses, port numbers, the sorts of devices being attacked, the malware's executables and instructions, and more [27]. In the field of computer security, honey pots have become more useful for analyzing malware and its variants. Fred Cohen's "The Deception Toolkit" It was created in 1998 [28] to stop malicious software that replicates itself, called "worms," and was made available to the public and sold soon after. These days, you can get Honey pots in a broad range of tastes, each optimized for a certain set of responsibilities. The level of help needed from the attacker is one way to classify it. There will be more involvement if a lot of data needs to be collected. Therefore, there are two distinct categories of honey pots, low- and high-interaction versions [9]. The purpose of a honey pot may be classified as either "Real-Time Honey pots" to protect a company's assets in real time against attacks or "Research Honey pots" to investigate possible threats and system weaknesses. Because of this, honey pots are effective in preventing Zero-Day DDoS Attacks without disrupting IoT [29].

The traditional honey pot is not the same as the Iota honey pot. Due to the large range of IoT devices, the architectures of IoT honey pots differ significantly from the consistent designs of classic honey pots (mainly x86 and x86-84). Our proposed solution includes a honey pot structure that has successfully captured many attempts to install malware onto the IoT device. When information is gathered, it is recorded. Our training machine learning model accepts file inputs. The most important benefit of using a honey pot to train a model rather than an already-existing dataset is that it allows the model to be trained on previously-undiscovered variants of malware families [13].

Our solution automates the detection and prediction of incoming security concerns to IoT devices via the application of machine learning, namely the deployment of relevant learning algorithms and methodology [17]. There are two primary categories of learning algorithms: supervised and unsupervised. Labels must be assigned during training for supervised learning to function, and then comparable attributes may be used to predict the same label. As opposed to

relying on labels, unsupervised learning [6] determines classifications based on commonalities seen in the training sample, rather than on labels. Since we like to avoid involving a human in the process—a specialist must set the rules and give the relevant labels—an unsupervised learning approach is preferred. Unsupervised learning encompasses a wide variety of methods, some of the most popular being cluster analysis, anomaly detection, and artificial neural networks. Malware detection might be seen as a classification or clustering problem [10, 11]. When there are already instances of the data, we may utilize supervised learning to make educated guesses about the nature of the categorization problem. Much of the clustering problem involves categorizing different types of unknown malware into groups based on their similar traits. Learning without the aid of a teacher [8]. It is suitable for identifying malware since machine learning creates less false positives and false negatives than other anomaly detection techniques [4].

II. RELATED WORK

There are a number of honey pot-based strategies for protecting against distributed denial of service attacks (DDoS) in the existing literature. Signature matching has previously been utilized as a foundation for detection in ways like these [16]. Signatures are used to identify malicious software, and the log files created by the honey pot are a primary source of these signatures [18]. This method of detection was limited in that it could only handle recognized malware families with stored signatures and their variants. Anomaly-based detection [12] is another option; instead of using rules, it establishes a threshold for typical user behavior and declares any variation from that as suspicious. Since attackers may now also replicate regular activity, such systems are prone to a high proportion of false positives. Also, because to its capacity for learning and teaching over time, machine learning based system is able to handle such a situation. Training the model using efficient and up-to-date data allows for more precise categorization with fewer false positives. Using the principles of machine learning, the ever-changing data collected by honey pots may be put to greater use, making future assaults more predictable.
For example, deep learning models like the Convolutional Neural Network (CNN) [22], the Recurrent Neural Network (RNN) [23], and others have been presented as machine learning based method to identify DDoS. (Recurrent Neural Network) [25], "Long Short-Term Memory Neural Network" [23], and "Gated Recurrent Unit Neural Network" [24]. A network-based anomaly detection approach was developed [26] that employs deep auto encoders to identify abnormal network traffic caused by hacked IoT devices by extracting behavior snapshots of the network. However, a lot of data is required for deep learning models to train themselves to provide reliable results. Still, they often take a long time to learn and have a training technique that is both difficult and computationally costly. Due to their limited resources and the need to provide services in real time, IoT devices cannot afford such elaborate processes. There is also a need to create new techniques for distinguishing between IoT-based assaults that last an hour and those that last a moment [26].

III. METHODOLOGY

Although malware detection is a primary focus of our proposed solution, we also want to uncover the identities of previously undiscovered malware families that fall under the categories of Distributed denial of service attacks that exploit newly discovered vulnerabilities. Because there are so many conceivable malware infection variations, a comprehensive DDoS protection against zero-day assaults cannot yet be developed [19]. This problem is addressed by using a honey pot strategy inside a machine learning based detection system. By luring in attackers on purpose, honey pots may record detailed information on the malware's characteristics and how it breaches IoT security [16]. Additionally, a machine learning based detection framework is used to predict the likelihood of abnormal activity based on the log files generated by the honey pot using a light weighted classification algorithm, ideally an unsupervised one as it does not require any expert to classify the training tulles into a malicious one or a normal one [20]. This is the design of our suggested solution: The procedure begins when an attacker tries to enter into an IoT device using various combinations of ID and Password in order to inject the malware via an open port (telnet port 23 or 2323). The honey pot comes into play because it allows the attacker to get past its defenses on purpose. The goal is to collect data about the intruder and the virus by keeping a log of all communications between the device and the intruder. The IP address, port number, and other details about the C&C server, as well as the nature of new malware families, variations, and targeted devices, are all captured in these log files. In order to train our machine learning model, we must now convert the data in our log files into a tabular format suitable for use as training datasets.

As a result, we'd rather not burden an application with a classification system that uses a lot of memory but requires as little training data as feasible to make accurate predictions. Interconnected electronic gadget [20]. Finally, action is taken that is suitable for the categorization result. The whole workflow of the suggested approach is shown in Fig.1. To make the process dynamic and readily unable on resource constrained IoT devices, training is repeated whenever the training data size limits are exceeded.

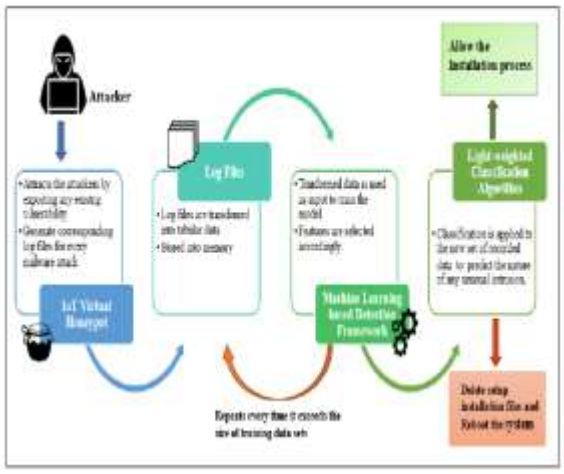

Fig. 1. Process flow for the honey pot-based solution with machine learning based detection framework.

IV. IMPLEMENTATION ASPECTS

Any new method or concept has to be put into practice before its viability and superiority over existing options can be assessed. As as was mentioned above, our suggested method entails a number of additional phases. Each stage allows us to include the most recent approaches to the underlying idea, ensuring that our solution is always cutting-edge enough to meet today's Internet of Things concerns. The two most crucial components in our strategy for achieving the required implementation are real-time machine learning detection and IoT honey pots, both of which have seen significant advancements in recent years.

IoT Cyber-Honey pot:

The first stage of our suggested strategy is to entice attackers into knowingly abusing the vulnerability in IoT devices. To simulate such actions, we require a system or device that can convincingly pose as an exploitable Internet of Things (IoT) device, hence convincing an attacker to carry out his malicious plan without questioning the authenticity of the vulnerabilities. IoT honeypots are the colloquial name for systems like this. As was said in the introduction, honey pots may be broken down into three distinct categories: high-interaction honey pots (HIH), low-interaction honey pots (LIH), and medium-interaction honey pots (MIH), which combine the characteristics of the first two. For IoT devices with limited resources, a high interaction honey pot (HIH) is impractical; hence a medium interaction honey pot (MIH) is the better choice. That's why we're calling it a "virtual" IoT honey pot rather than a "real" one: since we'll be deploying it digitally, by imitating the Iota platform using Iota communication protocols. As a result, the honey pot is able to record the attacker's methods of attack, including things like network traffic, payload, malware samples, toolkit, etc.

Recently developed Internet of Things honey pots for Does detection are listed below.

Hotpot [32] is a honey pot that, like others in the field, simulates the Telnet services of numerous Internet of Things devices via the cooperation of a frontend low interaction responder and a backend high interaction responder.IoTBOX is an interoperable virtual environment for interaction between devices that may run on a wide variety of CPU types.

• Telnet Iota honey pot [30]: The trap for Iota is implemented via a Telnet server. This TR-069 (CPE WAN Management Protocol)-specific honey pot, known as Honey Thing [31], simulates a susceptible modem/router (with an embedded web server running Rampage). Dionaea [33] is a honey pot that mimics the actions of Internet of Things devices by use of the MQTT protocol. Honey pots come in a variety of forms, and one that pretends to be a

ZigBee gateway is the ZigBee Honeypot [34]. This IoT honey pot is designed to catch hackers using Telnet, SSH, HTTP, and CWMP.

Thing Pot [29]: Unlike traditional IoT honey pots, which only mimic one layer of communication protocol, Thing Pot is able to simulate a whole IoT platform (e.g., Telnet, HTTP, etc.).The ideal IoT honey pot would be able to imitate not just the communication protocols used by the target IoT devices, but also the whole IoT platform and any supporting application layer protocols. IBM's Message Queue Telemetry Transport (MQTT), XMPP (Extensible Messaging and Presence Protocol) with its foundational support for instant messaging (IM) and presence, and others are among the most widely used application protocols for IoT connectivity.

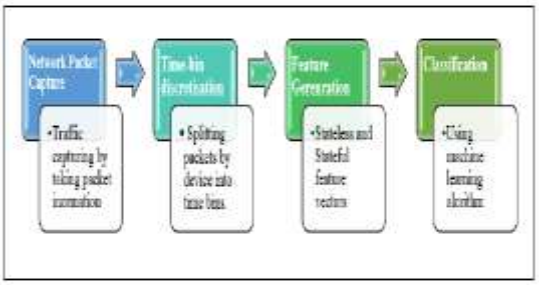

Figure 2: Detection framework process flow based on machine learning.

Protocols, such as the functionally-rich AMQP (Advanced Message Queuing Protocol) of the banking sector, the constrained-function Coop (Constrained-Scope Application Protocol), and the widely-used Protocol) created for low-power devices, the Universal Plug and Play suite of protocols for discovering networked devices, and the Hypertext Transfer Protocol (Representational State Transfer), often known as HTTP REST. When it comes to M2M communications and Internet of Things (Iota) systems, REST is the architectural style of choice. Thing Pot, among all the aforementioned honey pots, is suitable for our goal since it allows for a fascinating variety of potential malware assaults.

Machine Learning Detection Framework in Real Time

A crucial part of our Dodos detection method is a machine learning-based detection system. Several machine learning methods can do the necessary classification. We're looking for a machine learning solution that can categories the malware characteristics effectively and without producing a large number of false positives, and we need it to work in real time. For instance, R. Dashy et al., 2018 [17] recently provided a method for real-time machine learning based detection in Iota devices, which has been shown to identify malware with an accuracy of 0.99. As the number of Internet of Things (Iota) bonnet assaults has skyrocketed in recent years, our solution is designed with them in mind. Because Iota devices often connect with nearby endpoints rather than distant web servers, Iota traffic has certain characteristics that are not shared by ordinary laptop and smart phone traffic. Such patterns in Iota traffic may be studied in detail using a machine learning procedure. Data gathering, feature extraction, and binary classification are just a few of the procedures involved. Network flow parameters including packet length, inter-packet intervals, and protocol are among the most prominent facts gleaned from Iota-related networks. Random forests, K-nearest neighbors, support vector machines, decision trees, and neural networks are only some of the attack detection classifiers that are evaluated and compared. Effective classifiers include the random forest, K-nearest neighbors, and neural nets [17]. With the help of various machine learning algorithms, such as neural networks, it is possible to detect Dodos in Iota network traffic with greater accuracy by employing feature selection based on Iota-specific network behaviors, such as the small number of endpoints and the consistent time interval between packets.

Beginning with Traffic Capture, then Packet Grouping by Device and Time, and finally Feature Analysis, Anomaly Detection is a multi-step process. Phase of extraction, followed by the binary classification stage. All IP packets transmitted from an Iota device as part of a smart home application will have their timestamps, packet sizes, origin IP addresses, and destination IP addresses recorded as part of the traffic capture process. Due to the complexity and security hazards involved, gathering Dodos traffic is a difficult undertaking. TCP SYN flood, UDP flood, and HTTP GET flood simulations have been included to catch any future changes to malware characteristics.

Packets from Iota devices are sorted into groups by source IP address and then further subdivided into time stamps that do not overlap. Depending on how the connected device is behaving, the feature extraction procedure will create either stateless or tasteful features for each packet. Rather of separating incoming data based on its IP address,

stateless features are created based on characteristics shared by all packets in a given flow. As opposed to this, tasteful features focus on collecting data on the aggregated flows in the network traffic over relatively short intervals of time. Stateful characteristics include things like bandwidth and the uniqueness or cardinality of IP addresses, whereas stateless features include things like packet size and Inter-packet interval. Either way, I'm relieved.
Classification methods such as K-nearest neighbours, random forests, and support vector machines are used to perform binary classification. It's important to be able to tell Dodos activity from from regular traffic, thus researchers have turned to support vector machines and deep neural networks [36]. The whole sequence of events is shown in Fig. 2. Using deep learning classifiers is also advantageous because of the extra data they can analyze thanks to being put to use in real-world deployments.

V. CONCLUSION

The Internet of Things is the primary driver of the technological progress that has taken place in the physical world. It's the primary driver of cyber attacks, but it also has some negative consequences. Assaults, distributed denial-of-service attacks in particular. Because of this, protecting against attacks that use IoT to compromise networks is now the top priority in the area of Internet security. Some security techniques have been presented in the relevant area to make the IoT network resistant to these kinds of assaults; however as IoT bonnet attacks evolve, these defenses become obsolete. To combat Dodos attacks, we developed a honey pot-based system that employs a machine learning detection framework in real time. In order for ML-based detection frameworks to train their classifiers accurately, honey pots must be used to assure the tracking of newly emerging malware traits. We need to take this method to the next level, where we can use it on real-world situations to identify unresolved problems, so that it may be used in the future. In addition, a cloud server may be used to manage very low-powered Iota gadgets. To conclude, we may evaluate our solution's performance in light of that of competing models and draw conclusions from the results.

REFERENCES

Journal of Hardware and Systems Security, Volume 2: Issue 2 (2018), Pages 97-110, "Internet-of-Things Security and Vulnerabilities: Taxonomy, Challenges, and Practice" by K. Chen, S. Zhang, Z. Limy Zhang, Q.Deng, Sandip Ray, and Year Jin.
According to [2] "The Effect of Iota New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," published in IEEE Internet of Things Journal in 2018, W. Zhou, Y. Jiao, A. Pang, Y. Zhang, and P. Liu.
Source: [3] "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet of Things Journal, volume 4, issue 5, pages 1125-1142 (2017). Authors: J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao.

Using honeypots with the IoT [4]. Located at this link: https://securelist.com/honeypots-and-the-internet-of-things/78751.

Unsupervised learning. Hastie, T.; Tibshirani, R.; and Friedman, J. [5]. In Fundamentals of studying statistics (pp. New York: Springer, 2009.

"DDoS in the IoT: Mirai and Other Botnets," by C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, published in Computer, volume 50, issue 7 (2017), pages 80-84.

Dougherty, J., R. Kohavi, and M. Sahami. Discretization of continuous features, both supervised and unsupervised. Proceeding of the 1995 Conference on Machine Learning, pages 194-202 (1995).

According to [8] Sommer and Paxson (May 2010). Machine learning beyond the sandbox: detecting network intrusions using AI. 305–316), IEEE Symposium on Security and Privacy (SP). IEEE (2010).

Reference: [9] M. Anirudh, S. A. Thileeban, and D. J. Nallathambi, "Use of Honeypots for Mitigating DoS Attack Targeted on IoT Networks," 2017 International Conference On Computer, Communication, And Signal Processing (ICCCSP), Chennai, Pp. 1-4, (2017).

(2008, July) [10] Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. Malware categorization and learning. Detecting Intrusions, Malware, and Assessing Vulnerability: Proceedings of the International Conference (pp. 108-125). Publishing house Springer in the German cities of Berlin and Heidelberg.

Automated categorization and analysis of internet malware. [11] Bailey, M., Overhead, J., Andersen, J., Mao, Z. M., Bahamian, F., & Mazarin, J. Springer, Berlin, Heidelberg, pp. 178-197 in International Workshop on Recent Advances in Intrusion Detection (2007).

An Algorithm for Anomaly-based Botnet Detection, SRUTI, 6, 7-7. (2006), Binkley, J. R., and Singh, S.

[13] U.S. Patent No. 8,844,033, Song, Keromytis, and Solo. The United States Patent & Trademark Office (2014). Washington, DC.