ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org

Vol 9, Issuse.3 Aug 2021

Honey Pots and Machine Learning: A Defense Framework for Internet of Things Botnet DDoS Attacks

DR SASANKO SEKHAR GANTAYAT

.

Abstract—

Recent years have seen a meteoric rise in IoT botnet DDoS assaults, making IoT security a major priority for network administrators everywhere. Quite a few a numbers of security strategies have been offered for the Internet of Things, but all fall short when it comes to defending against the constantly evolving Zero-Day Attacks that are becoming more common. This study introduces a honey pot-based method for malware detection that makes use of machine learning. Iota honey pot data is utilized to efficiently and dynamically train a machine learning model. This method may serve as a good jumping off point for protecting the Internet of Things (Iota) against Dodos attacks, which have recently become a major threat.

I. INTRODUCTION

A distributed denial of service (Dodos) attack may now originate from the Internet of Items (Iota), which is a network of linked things that operates independently of human intervention [1]. The convenience of Internet of Things devices allows for more desktop PCs are more vulnerable to breaches. Thus, there has been a dramatic rise in bonnet assaults that use the Internet of Things [7]. Malware infestations in an Iota network lead to the creation of a bonnet, or a network of bots (compromised IoT devices) [2]. A recent analysis found that there are more than 6 billion IoT devices in use worldwide; with so many potentially susceptible devices, hackers can't afford to ignore the sector. There have been Hundreds of malware detections throughout the years, with 2017 accounting for over half of them [5]. By recording information about the attacking agent, such as malware for a DDoS assault, a honeypot may be used to watch and analyze the attacker's manner of initiating the attack [9]. It may mimic any weakness that can be readily exploited by an attacker, allowing it to become compromised on behalf of the primary server. IP addresses, MAC addresses, port numbers, the types of devices targeted, the malware executables and instructions, etc. may all be gleaned via monitoring the traffic between the attacker and itself [27]. Honey pots have proven to be an invaluable tool for studying malware and its variations in the area of computer security in recent years. The Deception Toolkit, created by Fred Cohen

PROFESSOR, Mtech,Ph.D
Department of CSE
Gandhi Institute for Technology,Bhubaneswar.

In 1998 [28] and released to the public and for sale shortly afterwards, was designed to combat self-replicating computer programmers known as worms. Today, Honey pots come in a wide variety of flavors, each best suited to a certain set of tasks. It may be categorized in terms of how much cooperation it requires from the attacker. If a lot of information has to be gathered, the degree of engagement will increase. As a result, there are two types of honey pots: those with little interaction and those with high interaction [9]. You may categories honey pots based on their intended use: either to safeguard a company's assets in real time against assaults so as to enhance overall security, or to conduct research into potential threats and system flaws (in which case, they'd be dubbed "Research Honey pots"). That's why honey pots are so useful for stopping Zero-Day DDoS Attacks without affecting the Internet of Things [29].

There is a distinction, however, between the classic honey pot and the Iota honey pot. The designs of traditional honey pots (mostly x86 and x86-84) are uniform, while the architectures of IoT honey pots are diverse because of the wide variety of Iota devices. Several attempts to implant malware onto the IoT device have been caught using a honey pot framework, which we have included as part of our suggested solution. Data is logged when it is collected. The machine learning model we're using for training can take files as input. The use of a honey pot for model training has many advantages over utilizing preexisting datasets, the most significant of which is the ability to train the model on undiscovered variants of malware families [13].

Our approach uses machine learning, namely the deployment of suitable learning algorithms and methodologies [17], to automate the process of detecting and predicting incoming security risks to IoT devices. In the world of learning algorithms, the two main types are supervised and unsupervised. For supervised learning to work, labels must be assigned during training so that similar characteristics may be used to predict the same label. In contrast, unsupervised learning [6] does not rely on predetermined labels; rather, it makes classifications based on shared characteristics in the training dataset. Since we do not want to include a person in the process—an expert is required to define the rules and provide the appropriate labels—we favor using an unsupervised learning method. Cluster analysis, anomaly detection, and artificial neural networks are three of the most used unsupervised learning techniques. Detecting malware may be thought of as a classification or clustering challenge [10, 11]. Supervised learning is used to make predictions about the nature of a classification issue when there are known examples of the data. Clustering unknown malware kinds into groups according to shared characteristics is a major part of the clustering issue. Using a method of learning without human supervision [8]. When compared to other anomaly detection approaches [4], machine learning's main benefit is that it produces less false positives and false negatives, making it ideal for the identification of malware.

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.

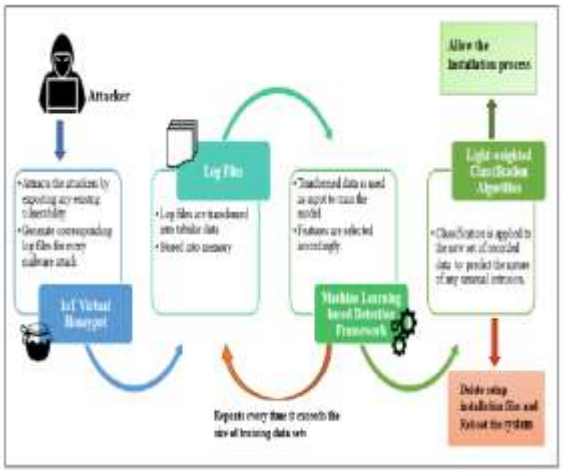

Fig. 1. Process flow for the honey pot-based solution with machine learning based detection framework.

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.

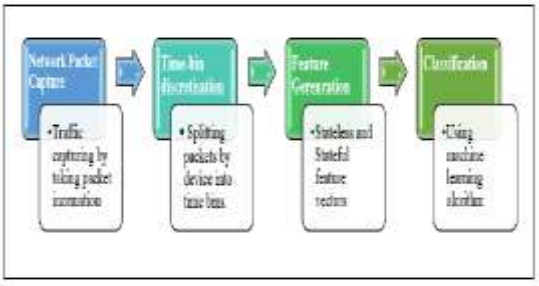

Figure 2: Detection framework process flow based on machine learning.

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

[1] K. Chen, S. Zhang, Z. Limy Zhang, Q.Deng, Sandip Ray, Year Jin, "Internet-of-Things Security and Vulnerabilities: Taxonomy, Challenges, and Practice" Journal of Hardware and Systems Security, vol. 2, Issue 2, pp. 97–110, (2018).

[2] W. Zhou, Y. Jiao, A. Pang, Y. Zhang and P. Liu, "The Effect of Iota New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," IEEE Internet of Things Journal. 2018.

[3] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1125-1142 (2017).

[4] Honey pots and the Internet of Things. Available at https://securelist.com/honeypots-and-the-internet-of-things/78751.

[5] Hastie, T., Tibshirani, R., & Friedman, J. Unsupervised learning. In The elements of statistical learning (pp. 485-585). Springer, New York, NY (2009).

[6] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," in Computer, vol. 50, no. 7, pp. 80-84 (2017).

[7] Dougherty, J., Kohavi, R., & Sahami, M. Supervised and unsupervised discretization of continuous features. In Machine Learning Proceedings 1995, pp.194-202 (1995).

[8] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), IEEE Symposium on (pp. 305-316). IEEE (2010).

[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).

[10] Rieck, K., Holz, T., Willems, C., Düssel, P., & Laskov, P. (2008, July). Learning and classification of malware behavior. In International Conference on

Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 108-125). Springer, Berlin, Heidelberg.
[11] Bailey, M., Overhead, J., Andersen, J., Mao, Z. M., Bahamian, F., & Mazarin, J. Automated classification and analysis of internet malware. In International Workshop on Recent Advances in Intrusion Detection Springer, Berlin, Heidelberg, pp. 178-197 (2007).
[12] Binkley, J. R., & Singh, S. An Algorithm for Anomaly-based Botnet Detection. SRUTI, 6, 7-7. (2006).
[13] Song, Y., Keromytis, A. D., & Solo, S. J. U.S. Patent No. 8,844,033. Washington, DC: U.S. Patent and Trademark Office. (2014).