International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 8, August 2020||

Internet of Things (IOT) Applications & Security: A Survey

Prof. Nanda Kulkarni

Assistant Professor, Department of Electronics & Telecommunication, Siddhant college of Engineering, Pune, India

ABSTRACT: Internet of things is going to lead the century with advent of 5G services.IoT system contain collection of sensors and devices. Because of an absence of security configuration just as the particular qualities of IoT gadgets, for example, the heterogeneity of processor engineering, IoT malware identification needs to manage exceptionally one of a kind difficulties, particularly on distinguishing cross-design IoT malware. In this manner, the IoT malware identification area is the focal point of exploration by the security network lately. There are numerous investigations exploiting notable dynamic or static examination for identifying IoT malware; be that as it may, static-based techniques are more compelling while tending to the multi-design issue. In this paper, we give an intensive study of static IoT malware location. We initially present the definition, advancement and security dangers of IoT malware. At that point, we sum up, look at and break down existing IoT malware discovery techniques proposed lately. At long last, we complete precisely the strategies for existing examinations dependent on the equivalent IoT malware dataset and a test design to assess impartially and expanding the unwavering quality of these investigations in recognizing IoT malware.

KEYWORDS*:* Internet of Things (IoT), Static-based, IoT botnet malware. Survey, IOT application

I. INTRODUCTION

Internet of Things (IoT) is the largest digital mega-trend that bridges physical and virtual worlds. The increment in the connectivity of people, objects, machines and the Internet is leading to the emergence of new business models as well as new interactions between mankind and the rest of the world. Due to the complexity of design and implementation in both hardware and software, as well as the lack of security functions and abilities, IoT devices are becoming an attractive target for cyber criminals who take advantage of weak authentication, outdated firmwares, and malwares to compromise IoT devices. By 2020, it is estimated that 25% of all cyber-attacks target IoT devices(https://www.gartner.com/). With the rapid adoption of IoT technologies in the industry, there will be an endless rise in these attacks. One of the most dangerous threats to IoT devices among them is malware. In Oct. 2016, Dyn (major US DNS service provider) was under one of the largest and most powerful DDoS attacks in recent history by Mirai malware family. The malware infected over 1.2 million IoT devices and targeted many popular online services such as Google, Amazon, etc.

Therefore, improving security aspects of IoT devices is becoming more and more urgent for researchers, especially when dealing with IoT malware. There are many research studies on security issues for IoT devices. Typical of such is Granjal et al. [1] who focused on analyzing extant protocols and mechanisms to secure communications for the IoT. Djamel Eddine Kouicem et al. [2] presented a comprehensive top down survey of the most existing proposed security and privacy solution in IoT. While, Djamel Eddine Kouicem et al. have categorized the various applications of IoT to identify security requirements and challenges for them, thereby analyzing traditional encryption solutions to deal with confidentiality, privacy, and availability. Besides, they also reviewed emerging technologies such as Blockchain and Software Defined Networking. In this perspective, a recent survey by Imran Makhdoom et al. [3] is quite comprehensive when presenting security issues and risks of threats to IoT devices. Besides, they emphasized that inherent safety provided by the communication protocols does not guard against harmful IoT malware and node compromise attacks. Hassan et al. [4] conducted a survey on security issues for IoT devices. However, the authors only focus on introducing solutions including lightweight authentication and encryption, not the IoT malware detection problem. Additionally, Felt et al. [5] reviewed the 46 pieces of mobile malware in the wild and collected dataset to evaluate the effectiveness of mobile malware identification and prevention methods. Costin et al. [6] only presented a comprehensive survey and analysis of all currently known IoT malware classes, without discussing IoT malware detection approaches.

IoT malware detection approaches could be classified into two main domains based on the type of strategy: dynamic and static analysis. Dynamic approach [7] consists of monitoring executables during run-time period and detecting abnormal behaviors. However, monitoring executing processes is resource-intensive, and in some cases,

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 8, August 2020||

malware could infect real environments. Besides, during execution time, it is not possible to fully monitor all their behaviors because many types of malware requiretrigger conditions to perform malicious behaviors. In addition to the common limitations of dynamic analysis, the execution of IoT executable files faces many issues such as diverse architectures (e.g., MISP. ARM, PowerPC, Sparc), and resource constraints of IoT devices. Therefore, it is difficult to configure an environment that meets the requirements for IoT executables to function correctly and fully. By contrast, static approach is performed by analyzing and detecting malicious files without executing them. One of the major advantages of static analysis is the ability to observe the structure of malware. In other words, we can explore all possible execution paths in malware sample without considering the diversity of processor architecture, thus making this approach effective to solve the heterogeneous issues of IoT devices. Thus, although there are many studies on security issues for IoT surveys, especially IoT malware detection, but no research has focused on methods of detecting IoT malware based on static analysis. Different from the existing survey studies when only evaluated based on the published results of the studies, this paper experimented exactly these methods with the same large dataset and the same system configuration.

In summary, the major contributions of this paper can be summarized as follows:

− First, we present a comprehensive overview of the interaction between the currently known IoT botnet families.

– Second, we provide a detailed taxonomy of static-features based IoT malware detection and discuss their shortcomings.

– Third, we accurately re-experimented the existing studies based on static analysis with the same large dataset and system configuration.

The rest of the paper is organized as follows: Section 2 provides an overview of IoT malware and its life-cycle, thereby better understanding the features that can be used by static analysis methods introduced in Section 3. Section 3 discusses the current IoT malware detection methods based on static features with their advantages and disadvantages. Section 4 presents and evaluates the methods by experiments. Furthermore, Section 5 shows the conclusion of this survey.

II. IOT MALWARE

In the past few decades, most malwares were designed to target personal computers running Microsoft Windows, the most popular operating system in the world with 83 percent of market share [8]. However, the diversity of computing devices has rapidly changed in recent years due to the Internet of Things technology. IoT devices are built upon a variety of CPU architectures, even on resource-constrained hardware such as Unix-based operating systems. Along with this change, IoT devices are becoming a favorite target by the attackers due to a lack of security design or implementation. In general, IoT malware has several characteristics such as IoT malware is used to perform DDoS attacks; IoT malware scans the open port of IoT services such as FTP, SSH or Telnet; IoT malware performs a brute-force attack to gain access to IoT devices.

Alex et al. [9] stated that most of the current malware generated by copying the source code according to online instructions or a variant of the same malicious code created by the malware writer. Through analysis, evaluation and synthesis of several studies such as [6], [10], [11] and manual analysis of some IoT malware samples this paper gives a brief chart of the recent development and evolution of IoT malware. Because IoT includes a vast and ever growing array of connected devices (e.g., smart meters, medical devices, public safety sensors, etc.), many IoT malware families such as Aidra, Bashlite and Mirai can utilize scanners that are designed to locate exposed ports and default credentials on these devices. In the last decade, IoT malware keeps developing and targeting new victims with various architecture. Mirai's evolution is gravitated toward changes in enterprise IT operations, extending its attack surface and bringing new zero-day exploits to consumer-level devices. In March 2019, IBM Xforce found a Mirai-like malware aimed at enterprises' IoT devices. These attacks drop crypto currency miners and backdoors onto affected devices.

There is a close correlation between IoT malware families reflected by the similarity of their functions as well as their source codes. Linux.Hydra as the first DdoS capable IoT malware that appeared since 2008. Since the release of Linux.Hydra's code, IoT malware developers have evolved into variants of Psybot, Chuck Norris and the last variation, Tsunami. However, part of the Tsunami's code was developed into the Remaiten and LightAidra, which are among the newest IoT malware. Also, the Tsunami is the ancestor of Bashlite and from Bashlite, the Mirai malware inherited and evolved more and more complex in 2016. From there, Mirai has continued to develop through variations that make it as a malware family rather than an individual stream of malware such as BrickerBot, VPNFilter. Accordingly, it cannot be denied that today's popularity of DDoS-capable IoT malware is steadily increasing because malware authors will continue to apply their creativity and programming skills to mutatetheir malwares for more critical infection on IoT devices. In summary, based on the analysis of the characteristics, evolution of IoT malware, we have found that there

are existing static characteristics of IoT malware that could be used as the features to detect malicious code, such as elf structure, strings, function call graph, grayscale image, etc. These features will be discussed in detail in Section 3

III. IOT MALWARE DETECTION BASED-ON STATIC FEATURE ANALYSIS

In this section, we discuss the static IoT malware detection methods that have been proposed since 2013. In static analysis, existing studies [7], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] commonly used the following static characteristics: control flow graph (CFG), operation codes (opcode), strings, file header, gray-scale image, etc. The way these features were extracted and processed greatly affects the accuracy and complexity of IoT malware detection methods. In the next subsection, we will demonstrate our method to divide these static-based features:

1) 3.1. Non graph-based IoT malware detection methods

Non graph-based detection methods aim at building a model that contains the properties of binary file structure to classify whether a binary file is malicious or benign. These methods rely on extracting the static features such as Operation Codes, Strings or File Structure to distinguish malicious samples. These characteristics can be divided into two groups: high-level features and low-level features. In particular, the low-level features can be obtained directly from binary file structure itself, while the high-level features must be extracted by disassembler (e.g., IDA Pro, radare2).

3.1.1. High-level features

Operation Code (Opcode) is one of the most popular features for malware detection. An Opcode is a single instruction that can be executed by the processor (CPU), which describes the behaviors of an executable file. In assembly language, an opcode is a command such as CALL, ADD or MOV. Based on this feature, Hamed HaddadPajouh et al. [12] proposed a method using Recurrent Neural Network (RNN) deep learning to detect IoT malware using the sequences of Opcodes. This method achieved the accuracy of 98.18 percent with the dataset of 281 ARM-based IoT malicious and 270 ARM-based IoT benign samples. Ensieh Modiri Dovom et al. [13] converted the executable files' Opcodes into a vector space and applied the fuzzy and fast fuzzy pattern tree methods to detect IoT malware. To prove this method in malware detection, they conducted an experiment on an ARM-based IoT dataset, comprising 1078 benign and 128 malware samples, and the experimental results achieved an accuracy of 99.83%. In similarity, Hamid Darabian et al. [14] presented a sequential opcode-based technique for IoT malware detection. By counting the number of opcode repetitions in executable files, the authors found that some opcodes in malware samples have higher frequency of repetition than benign files. Their experimental result achieved 99% accuracy and F-measure in the detection of IoT malware from benign samples. Nghi Phu et al. [22] proposed a feature selection method to detect cross-architecture malware, called CFDVex. The experiment achieved good results for cross-architecture malware detection. The experimental results show that the method is positive when it can detect the IoT malware for the MIPS architecture samples with a 95,72% accuracy rate by only train Intel 80386 architecture samples.

Strings — A string in an executable file is a sequence of characters such as "gayfgt" that is usually stored in ASCII (1 byte per character) or Unicode (2 bytes per character) format. Each printable string within an executable file could extract valuable information such as IP address, URL to connect, etc., to determine whether an executable is malicious or not [23]. Mohannad Alhanahnah et al. [15] generated good signatures for multi-architecture IoT malware classification based on printable strings. For evaluations, they used two IoTPOT malware datasets with 5150 malware samples, and with this signature-based detection mechanism, their experimental results achieved 95.5% IoT malware detection rate

3.1.2. Low-level features

ELF file header — An ELF (Executable and Linkable format) file format contains lots of interesting information that can be used in the malware detection. Based on this context, Farrukh Shahzad and Muddassar Farooq [16] presented a Linux malware detection tool, called ELF-Miner. To demonstrate their method, the dataset comprising 709 ELF samples was used and achieved more than 99.9% detection accuracy with less than 0.1% false alarm rate. In another work, Jinrong Bai et al. [17] introduced a method that extracts system call information from the symbol table of ELF files, they applied four machine learning algorithms for Linux malware detection. Using a dataset consisting of 756 benign and 763 malware samples, the experimental results achieved more than 98% accuracy on detecting an ELF file is malicious or benign.

Grayscale images — A grayscale image is type of image that each pixel has a value in the range from 0 to 255. In the malware detection problem, the executable files are analyzed and converted into binary strings 0 and 1, then combining those binary values into 8-bit vector segments that represent hex value from 00 to FF. These vectors are eventually converted into image data with a pixel value range between 0 and 255, where 0 as black and 255 as white. In this perspective, Su et al. [18] proposed a lightweight solution to distinguish between IoT malware and benign samples by feeding these gray-scale images to convolutional neural network model for detection. Besides, files of any size will

be normalized to fit with a 64 × 64 grayscale image and its remaining content will be deleted if redundant or covered with zero values if missing. The experiments achieved 93.33% accuracy for detecting IoT malware.

2) 3.2. Graph-based IoT malware detection methods

The most popular feature for malware detection is the Control Flow Graph. A control flow graph is a directed graph that represents all the possible execution paths that can be taken during the program, where each vertex (node) is represented by a basic block and each directed edge represents a possible control flow between the basic blocks. Control flow information in two key forms (1) inter-procedural control flow and (2) intra-procedural control flow. The interprocedural control flow graph demonstrates the association between functions and procedures in an executable file as a single CFG. Meanwhile, the intra-procedural control flow graph is demonstrated as a set of control flow graphs with one graph for a procedure or a function. Hisham Alasmary et al. [19] used the control flow graph (CFG) as an abstract structure to highlight the similarities and differences between IoT and Android malware binaries. Experimental results on the dataset consisting of 2.874 IoT malware samples and 201 android malware samples showed that IoT malware is more likely to contain a lesser amount of nodes and edges than Android malware, and graph-theoretic features between the IoT and android malware CFGs have a major variation. Therefore, they demonstrate the usefulness of CFGs in detecting IoT malware and android malware. Continuing to improve and expand this research direction in detecting IoT malware, Hisham Alasmary et al. [20] showed an IoT malware detection method that utilized Control Flow Graphs (CFGs) by using 23 static features to represent the characteristic of the CFG of IoT malware. By using this simple approach, this work achieved the accuracy of 99.66 percent with the dataset of 6000 malware and benign samples.

Follow the same approach, Azmoodeh et al. [21] proposed a deep learning-based method to detect Internet Of Battlefield Things (IoBT) malware that is based on the Opcodesequence graph. They evaluated the proposed method with a dataset including 128 malwares and 1078 benign files then achieved accuracy and precision rate of 98.37 percent and 98.59 percent respectively. HT Nguyen et al. [7] proposed an IoT botnet detection method based-on tracking footprints leaving at the steps of the botnet life cycle. These footprints were displayed as Printable String Information (PSI) which are used in the programming phase of any program such as IP address, username/ password patterns. In this work, they defined a graph-based data structure called PSI-Graph to represent the life cycle behavior of IoT botnet. On the dataset of more than 10000 samples, this study achieved the accuracy of more than 98 percent. In comparison with other methods, this work shows a better result in both detection rate and classifying time. Based on the above literature review, we compare the static IoT malware detection methods in Table 1 regarding their features used for detection, classification algorithms as well as the weaknesses that could affect the performance of these methods.

Method	Features	Mechanism	Passive technique	Weakness
[12]	Opcode	Identify malicious code through the sequences of Opcode	Neural networks	Only for ARM-based samples
[13]	Opcode	Apply fuzzy pattern tree to detect malicious sample	Fuzzing	Only for ARM-based samples. The dataset is not had enough samples.
[14]	Opcode	Detect malware by analyzing Opcode frequency	Machine learning	Only for ARM-based samples.
[22]	Opcode	Detect malware by using Vex intermediate representation	Machine learning	Only experiment with MIPS-based samples
[15]	Strings	Generate signature to classify IoT malware	Clustering	Time consuming Only for 4 malware families
[16]	ELF header	Extract features from sections of a binary file to detect malware	Machine learning	The structure of binary file is easy to modify.
[18]	Grayscale Image	Represent binary sample as grayscale image to detect malicious code	Neural network	Lose the accuracy when obfuscation or encrypt technique was applied

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 8, August 2020||

Method	Features	Mechanism	Passive technique	Weakness
[20]	CFG (Function Call Graph)	Calculate 23 properties of CFG to separate malicious and benign samples	Machine learning, Neural network	Time consuming The defined properties is not correct.
[21]	Opcode graph	Construct Opcode graph as a type of CFG to detect malware	Graph theory	Only for ARM-based samples
[7]	PSI graph	Use the PSI-Graph extracted from function call graph to detect malware	Neural network	Transforming graphs into vector data is time consuming

IV. CONCLUSION

IoT security is turning into an inexorably earnest issue in guaranteeing the wellbeing of the Internet framework and private information. This paper gave a deep reaching survey of developed IoT security and static-based recognition strategies. We talked about the principle methods just as qualities and shortcomings in existing static IoT malware discovery. In short, IoT malware location strategies can be separated into two gatherings: non diagram based and chart based techniques. The non diagram based strategies can accomplish a decent outcome when identifying "basic" and "direct" malware without customization or muddling, yet conceivably loses precision when distinguishing inconspicuous malware. In actuality, the chart based strategies show points of interest while dissecting the control stream of IoT malware, in this way can possibly precisely identify inconspicuous or muddled malignant code in spite of the unpredictability of these techniques. To think about the exhibition of these investigations. In light of the component, location examination and preparing time, we summed up the focal points and confinements of these investigations that can be utilized to improve the proficiency in future explores. As further augmentation of this work, we intend to structure and build up a chart based lightweight recognition technique that will assist with managing recognize malevolent executable record in IoT gadgets.

REFERENCES

[1] Granjal J., Monteiro E., Silva J.S. Security for the internet of things: a survey of existing protocols and open research issues
IEEE Commun. Surv. Tutor., 17 (3) (2015), pp. 1294-1312
[2] Kouicem Djamel Eddine, Bouabdallah Abdelmadjid, Lakhlef HichamInternet of things security: a top-down surveyComput. Netw., 141 (2018), pp. 199-221
[3] Makhdoom Imran, et al.Anatomy of threats to the internet of thingsIEEE Commun. Surv. Tutor., 21 (2) (2018), pp. 1636-1675
[4] Hassan W.H., et al.Current research on internet of things (IoT) security: A surveyComput. Netw., 148 (2019), pp. 283-294
[5] Felt A.P., Finifter M., Chin E., Hanna S., Wagner D.A survey of mobile malware in the wildPresented at the Security and Privacy in Smartphones and Mobile Devices (SPSM) (2011), pp. 3-14
[6] Costin Andrei, Zaddach JonasIot malware: Comprehensive survey, analysis framework and case studiesBlackHat USA (2018)
[7] Nguyen H.T., Ngo Q.D., Le V.H.A novel graph-based approach for IoT botnet detectionInt. J. Inf. Secur. (2019), pp. 1-11
[8] Cozzi Emanuele, et al.Understanding linux malwareIEEE Symposium on Security and Privacy (2018), pp. 161-175
[9] Allix K., Jerome Q., Bissyande T.F., Klein J., State R., Traon Y.L.A forensic analysis of android malware – How is malware written and how it could be detected?Presented at the IEEE 38th Annual Computer Software and Applications Conference (2014), pp. 384-393
[10] Angrishi KishoreTurning internet of things (IoT) into internet of vulnerabilities (IoV): IoT botnets(2017)arXiv preprint arXiv:1702.03681
[11] De Donno Michele, Dragon Nicola, Giaretta AlbertoDDoS-Capable IoT malwares: Comparative analysis and mirai investigationSecurity and Communication Networks, Wiley (2018)

[12] HaddadPajouh Hamed, Dehghantanha Ali, Khayami Raouf, Choo Kim-Kwang RaymondA deep recurrent neural network based approach for internet of things malware threat huntingFuture Gener. Comput. Syst., 85 (2018), pp. 88-96
[13] Dovom Ensieh Modiri, et al.Fuzzy pattern tree for edge malware detection and categorization in IoT J. Syst. Archit. (2019)
[14] Darabian Hamid, et al.An opcode-based technique for polymorphic Internet of Things malware detection Concurr. Comput.: Pract. Exper. (2019)
[15] Alhanahnah Mohannad, Lin Qicheng, Yan QibenEfficient signature generation for classifying crossarchitecture IoT malware Commun. Netw. Secur. (2018), pp. 1-9
[16] Shahzad F., Farooq M.Elf-miner: Using structural knowledge and data mining methods to detect new (linux malicious executablesKnowl. Inf. Syst., 30 (3) (2012), pp. 589-612
[17] Bai Jinrong, Yang Y., Mu S., Ma Y. Malware detection through mining symbol table of Linux executables Inf. Technol. J., 12 (2) (2013), pp. 380-383
[18] Su Jiawei, Vargas Danilo Vasconcellos, Prasad Sanjiva, Sgandurra Daniele, Feng Yaokai, Sakurai Kouichi Lightweight classification of IoT malware based on image recognition 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Vol. 2 (2018), pp. 664-669
[19] Alashmary Hisham, et al. Graph-based comparison of IoT and android malware International Conference on Computational Social Networks (2018), pp. 259-272
[20] Alasmary Hisham, et al. Analyzing and detecting emerging internet of things malware: A graph-based approach IEEE Internet Things J., 6 (5) (2019), pp. 8977-8988
[21] Azmoodeh Amin, et al. Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning IEEE Trans. Sustain. Comput. (2018), pp. 88-95
[22] T.N. Phu, L.H. Hoang, N.N. Toan, N.D. Tho, N.N. Binh, CFDVex: A novel feature extraction method for detecting cross-architecture IoT malware, in: Proceedings of the Tenth International Symposium on Information and Communication Technology, 2019, pp. 248–254.
[23] Islam Rafiqul, et al. Classification of malware based on integrated static and dynamic features J. Netw. Comput. Appl. (2013), pp. 646-656
[24] Pa Y.M.P., Suzuki S., Yoshioka K., Matsumoto T., Kasama T., Rossow C. CIoTPOT: A novel honenypot for revealing current IoT threatsJ. Inf. Process., 24 (2016), pp. 522-533