Section A-Research paper
ISSN 2063-5346

DESIGN AND ANALYSIS ON A MALWARE DETECTION APPROACH FOR INTERNET OF BATTLEFIELD THINGS

Dr. B. Narendra Kumar
Professor, Department of IT
Sridevi Women's Engineering College
Telangana
swecnarendra@gmail.com

B. Sravani
BTech Student, Department of IT
Sridevi Women's Engineering College
Telangana
Sravanibattul30@gmail.com

P. Nikitha
BTech Student, Department of IT
Sridevi Women's Engineering College
Telangana
Nikitashetty440@gmail.com

S. Meghana
BTech Student, Department of IT
Sridevi Women's Engineering College
Telangana
meghasukkayapally@gmail.com

ABSTRACT

A class of malware detection approaches converts benign and malicious files into control flow graphs (CFG) for improved malware identification. This helps to increase the accuracy of malware detection on the Internet of Battlefield Things (IoBTs). During the building of CFG, disassemblers are used to convert the binary code of a file into opcodes. These opcodes are then used in the creation process. chance CFGs are constructed in which the vertices represent the opcodes and the edges between the opcodes reflect the chance of occurrence of those opcodes in the file. These probability CFGs can be used to analyze and predict the behavior of programs. The probabilistic conditional fuzzy graphs (CFGs) are input into the deep learning model so that it can undergo additional training and testing. The probability of CFGs is directly proportional to the accuracy of the deep learning model. The result of the deep learning malware detection model is likely to be more accurate if the graph creation approaches can reflectorize the binary file with a higher degree of precision. In this study, we highlight the limitations of the existing probability CFG techniques, suggest a new strategy for the generation of probability CFGs that is a combination of crisp and heuristic approaches and calls itself HeuCrip, and then compare the proposed technique with the existing state-of-the-art schemes. The findings of the experiments indicate that the HeuCrip obtained an

accuracy of 99.93% and demonstrates a considerable improvement in performance in comparison to other state-of-the-art methods currently in use

.

1. INTRODUCTION

The Internet of Things (IoT) is the most significant development in the realm of digital megatrends because it connects the real and virtual worlds. The increased connection of people, objects, machines, and the Internet is causing the formation of new business models as well as new interactions between humans and the rest of the world. This is due to the fact that these factors are all becoming more interconnected. IoT devices are becoming an attractive target for cybercriminals who take advantage of weak authentication, outdated firmwares, and malwares in order to compromise IoT devices because of the complexity of the design and implementation in both hardware and software. In addition, there is a lack of security functions and abilities, which makes IoT devices an attractive target. According to research found at https://www.gartner.com/, it is anticipated that Internet of Things (IoT) devices would be the target of 25% of all cyberattacks by the year 2020. There will be a never-ending increase in the number of assaults like this as a direct result of the increasing adoption of IoT technologies in the sector. Malware is one of the dangers that IoT devices face, and it is one of the most deadly. The Mirai virus family carried out one of the most extensive and potent distributed denial of service assaults in recent history against the major US DNS service provider Dyn in October of 2016. The malicious software infected more than 1.2 million Internet of Things devices and targeted numerous well-known internet businesses, including Google, Amazon, and others.

As a result, enhancing the security features of devices connected to the internet of things (IoT) is becoming an increasingly pressing matter for academics, particularly when dealing with malware connected to IoT. Numerous research studies have been conducted on the topic of security concerns for Internet of Things devices. Granjal et al. [1], who concentrated on assessing existing protocols and procedures to secure communications for the internet of things, are a good example of this type of research. Djamel Eddine Kouicem and colleagues [2] offered a top-down comprehensive evaluation of the most existing proposed security and privacy solutions for the Internet of Things (IoT). While Djamel Eddine Kouicem et al. have categorized the many applications of IoT in order to identify security requirements and obstacles for them, this article will analyze traditional encryption options in order to cope with confidentiality, privacy, and availability issues. In addition to that, they discussed up-and-coming technologies such as Blockchain and Software Defined Networking. When seen from this angle, a recent survey conducted by Imran Makhdoom and colleagues [3] is fairly exhaustive in its presentation of security concerns and the risks posed by threats to IoT devices. In addition, they underlined that the intrinsic safety given by the communication protocols does not protect

against malicious Internet of Things malware and node compromise assaults. Hassan et al. [4] carried out a survey to investigate the many concerns regarding the safety of Internet of Things devices. The authors, on the other hand, are solely concerned with presenting potential solutions, such as lightweight authentication and encryption, rather than the issue of IoT malware detection. In addition, Felt et al. [5] examined the 46 different types of mobile malware currently circulating in the wild and analyzed the data that they acquired in order to evaluate the efficiency of various strategies for identifying and preventing mobile malware. Costin et al. [6] only gave a detailed review and analysis of all of the already known IoT malware types, but they did not mention any IoT malware detection methodologies.

On the basis of the type of strategy employed, ways to detecting malware in IoT could be categorized into one of two primary domains: dynamic analysis or static analysis. The dynamic approach [7] is keeping an eye on executables as they are being run and looking for any unexpected behaviors they may exhibit. Monitoring processes that are actively running requires a significant investment of time and resources, and there is always the risk that malicious software will spread to live systems. In addition, it is not possible to fully monitor all of their activities while they are being executed. This is because many forms of malware rely on certain trigger situations in order to carry out their dangerous behaviors. In addition to the limits that are typically associated with dynamic analysis, the execution of executable files that are associated with IoT presents several challenges. These challenges include different architectures (for example, MISP, ARM, PowerPC, and Sparc), as well as the resource constraints that are associated with IoT devices. As a result, properly configuring an environment that satisfies the requirements for Internet of Things executables to operate correctly and completely might be challenging. A static technique, on the other hand, examines and identifies harmful files without actually running them. This process takes place in the background. The capability of static analysis to observe the structure of malware is one of the most significant advantages of using this technique. In other words, we are able to investigate any and all conceivable execution routes in the malware sample without taking into account the various processor architectures, which is one of the reasons why this technique is useful for resolving the difficulties posed by heterogeneous IoT devices. Therefore, even though there are numerous studies on security concerns for IoT surveys, particularly IoT malware detection, no research has focused on techniques of identifying IoT malware based on static analysis. This is despite the fact that there are many studies on security concerns for IoT surveys. This research experimented with exactly these methodologies using the same large dataset and the same system architecture, which differentiates it from the current survey studies when only evaluated based on the published outcomes of the studies.

2. IOT MALWARE

The vast majority of malicious software, or malware, has been developed over the past

few decades to attack personal computers running Microsoft Windows, which is the most widely used operating system in the world, with 83 percent of the market share [8]. The technology behind the Internet of Things has been responsible for the rapid expansion of the variety of computing devices available in recent years. IoT devices are constructed using a wide variety of CPU architectures; some of them even run on hardware with limited resources and use operating systems based on the Unix kernel. As a result of the absence of security that is either designed into them or implemented into them, Internet of Things devices are quickly becoming a preferred target for cybercriminals. IoT malware, in general, is characterized by a number of characteristics, some of which include the following: IoT malware can be used to carry out DDoS attacks; IoT malware can scan the open port of IoT services such as FTP, SSH, or Telnet; and IoT malware can carry out a brute-force attack to get access to IoT devices.

According to Alex et al. [9], the majority of the currently active malware is produced by copying the source code and following the available instructions for doing so, or it is a variant of the same malicious code that was written by the person who wrote the malware. This work provides a brief chart of the current development and evolution of IoT malware, as shown in Fig. 1. This chart was created by the investigation, evaluation, and synthesis of numerous studies, such as [6, [10], and [11], as well as the manual analysis of several IoT malware samples. Many IoT malware families, such as Aidra, Bashlite, and Mirai, are able to utilize scanners that are designed to locate exposed ports and default credentials on connected devices like smart meters, medical devices, and sensors for public safety. This is possible because the Internet of Things includes a vast and ever growing array of connected devices (for example, smart meters, medical devices, and sensors for public safety). IoT malware has continued to evolve over the past decade, and it now targets new victims using a wide variety of attack architectures. Mirai's development has been moving toward alterations in enterprise IT operations, which has increased the size of its attack surface and introduced new zero-day exploits to consumer-level devices. In March of 2019, IBM Xforce discovered malware similar to Mirai that targeted the Internet of Things devices used by businesses. These assaults install cryptocurrency miners and backdoors on the devices that have been compromised.

There is a strong connection between different families of Internet of Things malware, which is shown in the fact that their source codes and functions are very similar to one another. Figure 1 shows Linux.Hydra, which was the first DDoS-capable Internet of Things malware to be discovered after 2008. IoT malware developers have progressed into variations like Psybot, Chuck Norris, and Tsunami since the code for Linux.Hydra was made public. Tsunami is the most recent iteration. However, a portion of the code for the Tsunami was later included into the IoT malware families known as Remaiten and LightAidra, which are among the most

recent. In addition, the image demonstrates that Tsunami is the progenitor of Bashlite, and that the malware known as Mirai inherited and evolved from Bashlite in 2016, becoming increasingly complicated. From that point on, Mirai has continued to advance through a variety of iterations, making it more of a malware family than an isolated stream of malware like BrickerBot or VPNFilter. Malware authors will continue to apply their ingenuity and programming abilities to mutate their malwares for more critical infection on IoT devices. As a consequence of this, it is impossible to deny that the prevalence of IoT malware that is capable of launching a DDoS attack is constantly growing in today's world.

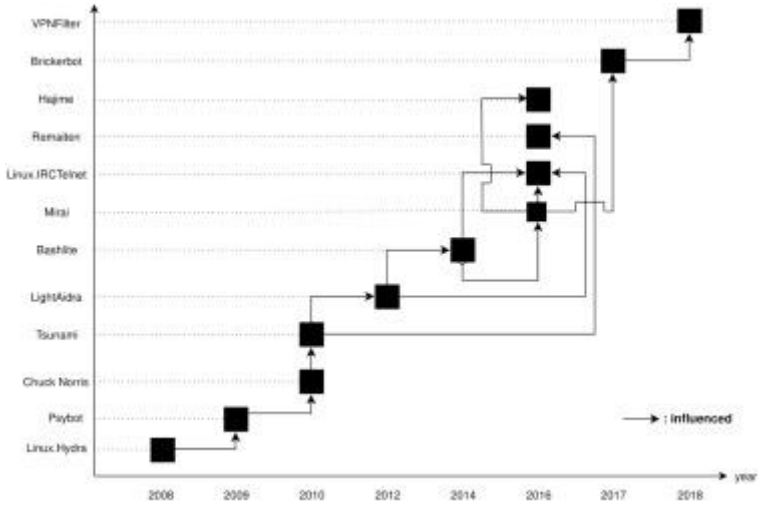

Fig. 1. The evolution of IoT malware.

3. IOT MALWARE DETECTION BASED-ON STATIC FEATURE ANALYSIS

In the following section, we will go over the static IoT malware detection approaches that have been proposed since 2013. Existing research frequently made use of the following static features while doing static analysis: control flow graphs (CFG), operation codes (opcode), strings, file headers, gray-scale images, and so on. The manner in which these features were retrieved and processed has a significant impact on both the precision and the level of difficulty of IoT malware detection approaches. In the following part, we will present our method for dividing these static-based features, which can be seen illustrated in Fig. 2.

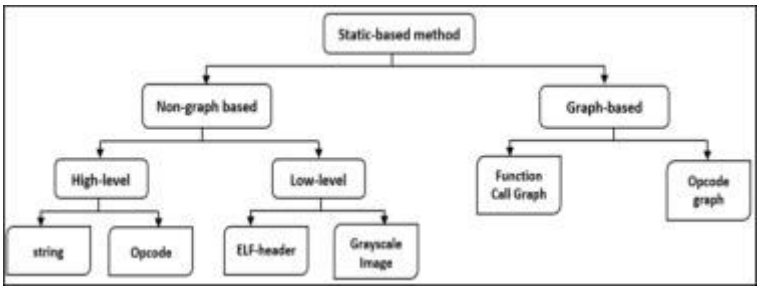

Fig. 2. Taxonomy of static-based features in IoT malware detection.

3.1. Non graph-based IoT malware detection methods

The goal of detection approaches that are not based on graphs is to construct a model that contains the characteristics of the structure of binary files in order to determine whether or not a binary file is harmful. In order to differentiate between malicious samples, these techniques involve the extraction of static information such as operation codes, strings, or file structures. These attributes are classified as either high-level or low-level features, depending on their degree of complexity. In example, the low-level characteristics can be acquired directly from the binary file structure itself, whereas the high-level features have to be extracted using a disassembler (for instance, IDA Pro or radare2).

3.1.1. High-level features

One of the most often used characteristics for identifying malware is known as the Operation Code, or Opcode. The behaviors of an executable file are characterized by a single instruction known as an Opcode. This instruction can be carried out by the central processing unit (CPU). An opcode is a command in assembly language. Some examples of opcodes include CALL, ADD, and MOV. Hamed HaddadPajouh et al. [12] suggested a method employing Recurrent Neural Network (RNN) deep learning to detect IoT malware using the sequences of Opcodes. This method was based on this characteristic. With the dataset consisting of 281 ARM-based Internet of Things malicious samples and 270 ARM-based Internet of Things benign samples, this approach achieved an accuracy of 98.18 percent. The authors Ensieh Modiri Dovom et al. [13] used the fuzzy and fast fuzzy pattern tree methods to identify IoT malware. They did this by converting the Opcodes of executable files into a vector space and applying those algorithms. In order to demonstrate the efficacy of this method in the identification of malware, they carried out an experiment on an ARM-based IoT dataset. The dataset contained 1078 benign and 128 malicious samples, and the results of the experiment obtained an accuracy of 99.83%. A sequential opcode-based method for detecting malware in internet-connected devices was presented by Hamid Darabian et al. [14] in a study that was comparable. The authors discovered, through the process of measuring the amount of opcode repeats in executable files, that some opcodes found in malware samples have a higher frequency of repetition than opcodes found in benign files. The results of their experiments showed a 99% accuracy and F-measure in the

differentiation of malicious Internet of Things samples from benign ones. CFDVex is the name of a feature selection algorithm that was proposed by Nghi Phu and colleagues [22] to identify cross-architecture malware. The experiment was successful in achieving its goals regarding the identification of malware across architectures. The results of the experiments indicate that the strategy is successful when it is able to identify IoT malware for samples of MIPS architecture with an accuracy rate of 95.72 percent using just Intel 80386 architecture samples for training.

3.1.2. Low-level features

ELF file header — The Executable and Linkable format, more often known as ELF, is a file format that stores a lot of relevant information that can be put to use in the detection of malware. ELF-Miner is a Linux malware detection tool that was presented by Farrukh Shahzad and Muddassar Farooq. The presentation was based on the scenario described above. For the purpose of demonstrating their method, they employed the dataset that had 709 ELF samples and achieved a detection accuracy of more than 99.9% while having a false alarm rate of less than 0.1%. In a different piece of work, Jinrong Bai and colleagues presented a method that extracts information about system calls from the symbol table of ELF files. They then used four different machine learning methods to the problem of detecting malware in Linux. The experimental results obtained an accuracy of more than 98% while attempting to determine if an ELF file is dangerous or benign by using a dataset that included 763 examples of malware and 756 samples of benign files.

Grayscale images — A pixel in a picture that is grayscale has a value that falls somewhere between 0 and 255, depending on the type of image. The executable files are evaluated and transformed into binary strings consisting of 0 and 1, and then those binary values are combined to form 8-bit vector segments that represent hex values ranging from 00 to FF. This is done as part of the malware detection challenge. In the end, these vectors are transformed into image data that has pixel values that vary from 0 to 255, with 0 representing black and 255 representing white. When seen from this angle, Su et al. offered a lightweight approach to differentiate between IoT malware and benign samples by feeding these gray-scale images to a convolutional neural network model for the purpose of detection. In addition, files of any size will be normalized to fit within a grayscale image that is 64 by 64 pixels, and the remaining content of the file will either be erased if it is unnecessary or covered with zero values if it is missing. The experiments were successful in identifying IoT malware with a rate of accuracy of 93.33%.

CONCLUSION

The elimination of Internet of Things malware is becoming an increasingly pressing concern in the context of protecting the integrity of the Internet infrastructure and personal information. In this study, a complete assessment of newly discovered pieces of malware targeting internet of things devices and static-based detection approaches was provided. We

talked about the most common methods, as well as the advantages and disadvantages of the currently available static IoT malware detection. In conclusion, Internet of Things virus detection approaches can be broken down into two categories: methods that do not use graphs and methods that do use graphs. The non-graph-based approaches have the potential to obtain a satisfactory result when identifying "simple" and "forthright" malware without customisation or obfuscation, but they may lose accuracy when detecting malware that has not been seen before. Graph-based methods, on the other hand, have advantages when evaluating the control flow of IoT malware. As a result, despite the complexity of these methods, they have the potential to accurately discover undetected or intricate harmful code. In order to compare the effectiveness of these research, we also put them into practice and analyzed them using the same Internet of Things dataset, which included 11200 individual samples. We analyzed the benefits and drawbacks of these investigations based on the mechanism, detection analysis, and processing time. This information may be utilized to improve the efficacy of future study by reducing wasted time and effort. As a further extension of our study, we intend to create and develop a graph-based lightweight detection approach that will assist with identifying harmful executable files in IoT devices. our will help us cope with the issue of distinguishing between legitimate and malicious files.

.

REFERENCES:

[1] E. Bertino, K.-K. R. Choo, D. Georgakopolous, and S. Nepal, "Internet of things (iot): Smart and secure service delivery," ACM Transactions on Internet Technology, vol. 16, no. 4, p. Article No. 22, 2016.

[2] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," Journal of Network and Computer Applications, 2017.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645– 1660, 2013.

[4] F. Leu, C. Ko, I. You, K.-K. R. Choo, and C.-L. Ho, "A smartphonebased wearable sensors for monitoring realtime physiological data," Computers & Electrical Engineering, 2017.

[5] M.Roopaei, P. Rad, and K.-K. R. Choo, "Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging," IEEE Cloud Computing, vol. 4, no. 1, pp. 10–15, 2017.

[6] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K. R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," Future Generation Computer Systems, 2017.

[7] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010