Computer Communications 147 (2019) 14–20

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Software-defined forensic framework for malware disaster management in Internet of Thing devices for extreme surveillance

Visu P. [a,*], Lakshmanan L. [b], Muruganantham V. [c], Meenaloshini Vimal Cruz [d]

[a] Department of CSE, Velammal Engineering College, Chennai, Tamilnadu, India
[b] Department of CSE, Sathyabama Institute of Science and Technology, Chennai, Tamilnadu, India
[c] SEEMIT, Institute Technology Pertama, Mantin, Negeri Sembilan, Malaysia
[d] Department of Computer Science, Keene State College, Keene, USA

ARTICLE INFO

Keywords:
Malware
Portable and executable files
Cybercrime
Malware disaster management
Extreme surveillance

ABSTRACT

Malware perception is an important technique which has to be explored to analyze the corpus amount of malware in short duration for effective disaster management. Accurate analyses of malware must be done by detecting them in initial stage in an automatic way to avoid severe damage in Internet of Thing devices. This is enabled by visualizing malware by using a software-defined visual analytic system. Though many auto analysis techniques are present visualization of malware is one of the effective techniques preferred for large analysis. Malware exhibits malicious behavior on computing devices by installing harmful software such as viruses. The existing static and dynamic form of malware detection is an inefficient technique as it involve in disassembling of malicious code. In this project, the visualization of malware in the form of images is proposed in order to find the malicious insertion on the executable files of computing devices for extreme surveillance. The malware detection becomes easier to visualize the malicious behavior in form of images by feature based classification of images as the global property of exe gray scale image is unchanged. This will be an eye open in healing the security issues in cyber-crime and provide extreme surveillance.

1. Introduction

Malware are the malicious software which often gain access to computer and cause damage to the in Internet of Thing (IoT) devices, without the knowledge of legitimate user. The characteristics of malware depend on their type. Analysis of malware states that, not all malwares are malicious. Some malware are designed for monitoring the data and stealing the content without causing damage to the system. Usually malware either acts as data stealer or some time cause severe damage to the system, like crashing the system in extreme stages. So it is important to block the malware and heal the existing one for extreme surveillance. Delayed detection of malware is some time useless. So, it is important to do reverse engineering to gain the access of data which are lost for extreme surveillance.

Enormous amount of malware is peeping out every day in IoT devices. So it is important to detect malware and analyze them. The most common method used is signature based and anomaly based detection. Due to increase in the malware and its variant, it is important to improvise the method of malware detection by working in speeding up of detection process [1]. To incorporate the method of increasing the analysis speed, visualization technique is used. Visualization based analysis help in obtaining the gist of malware and easier classification in time consuming manner. The classical approach like signature based detection and behavioral analysis method does not support detection of new emerging malware for extreme surveillance. Massive amount of samples have made the need for automated data analysis. To incorporate them, in analysis of system calls and there pattern have to be made for classifying in terms of malicious or non malicious. Static analysis offers the foremost complete coverage however it always suffers from code obfuscation. The practicable needs to be unpacked and decrypted before analysis, and even then, the analysis are often hindered by issues of wild quality. Dynamic analysis is additional economical and does not would like the practicable to be unpacked or decrypted. However, it is time intensive and resource overwhelming, therefore raising quantifiability problems. To improve the existing analysis approach visualization method is preferred.

The visualization of malware is made either by analyzing individual malware or can be made by analyzing the group of malware in IoT Devices [2,3]. Individual analysis of malware is easier then cluster analysis but they are not suitable for larger collection of malware. So to analysis enormous malware two important are widely preferred namely

[*] Corresponding author.
E-mail addresses: pandu.visu@gmail.com (Visu P.), laks14@yahoo.com (Lakshmanan L.), mail_muru@yahoo.com (Muruganantham V.), meenalosini.Vimal.Cruz@keene.edu (M.V. Cruz).

https://doi.org/10.1016/j.comcom.2019.08.013
Received 26 June 2019; Received in revised form 30 July 2019; Accepted 11 August 2019
Available online 13 August 2019

feature based and image based approach. Visualization tool mainly depend on data which are usually obtained either by static or dynamic analysis. The static analysis is made without actually executing the code. They just involved in checking the basic property like file type, strings, checksum etc. Analyzing the data after execution is called as dynamic analysis.

The analysis made by virtualization can be done either by physical, virtual or emulated format. The execution of samples is done directly on the main system, where the operation is installed is called physical analysis. They are not much secure because they are having same architect as host machine. The main disadvantage is malicious file can easily affect the hardware, as it is depended on it. So virtualization technique is widely preferred, as they help in interaction between host software and their supported hardware. In this technique, the data's are made to execute on the virtual machine monitoring for extreme surveillance. Other technique used is emulation type where the isolation is made for physical machine. The major drawback of emulation is when sandbox is detected; they gain full access to system and leads to severe problem. Still sandboxes are preferred for translating the states of register or CPU to the top level information like registry and file operation. Visualize the malicious behavior in form of images by feature based classification of images as the global property of exe gray scale image is unchanged. This will be an eye open in healing the security issues in cyber crime.

2. Related works

Visualization is done by analyzing the image of malware which are generated based on its behavior. The behavioral pattern is proposed by Lakshmanan Nataraj et al. [4], obtain behavioral feature and are executed in the virtual machine. To discriminate the varying behavioral pattern unique color is assigned to each behavior. Once the unique color to behavioral mapping is done, malware images are generated by clustering the behaviors. Visualization tool mainly depend on data which are usually obtained either by static or dynamic analysis. The static analysis is made without actually executing the code. They just involved in checking the basic property like file type, strings, checksum etc. Analyzing the data after execution is called as dynamic analysis. The analysis made by virtualization can be done either by physical, virtual or emulated format. The execution of samples is done directly on the main system, where the operation is installed is called physical analysis. They are not much secure because they are having same architect as host machine. The main disadvantage is malicious file can easily affect the hardware, as it is depended on it. So virtualization technique is widely preferred, as they help in interaction between host software and their supported hardware. In this technique, the data's are made to execute on the virtual machine monitoring. Other technique used is emulation type where the isolation is made for physical machine. The major drawback of emulation is when sandbox is detected; they gain full access to system and leads to severe problem. Still sandboxes are preferred for translating the states of register or CPU to the top level information like registry and file operation.

The behavioral features are obtained by executing the API in the virtual machine. This ensures that the main system is not affected by the malicious code execution. The accurate nature of malware can be obtained by analyzing the user mode API calls. Kernel mode API is usually not preferred for analyzing the behavior because it often results in flooding, which in turn will affect the behavioral image. Random allocation of color mapping is not preferred because they might not give meaning full information. So the behavioral are clustered based on the malicious range. This is done for easier analysis. As all the API calls are not malicious, it is important to classify them based on level of maliciousness. Based on the similarities in the behavior, the images can be further classified to find the malware variant. It is proposed by Conti, G. and Bratus [5].

The framework is developed by Trinius, P. Holz, [6] based on visualization to perform reverse engineering. By this technique they determine the functional unit and retrieve the obfuscated data by performing visualization. They visualize the state transition in the address and link. While the thread graph are used for representing the visualized result. The visual environment analyze of code is made to understand them better. The classification of various binary fragments is made based on the predetermined statistical approach [7]. Both static and dynamic analysis based clustering and categorizing of malware is made. Knote, et al. [8,9] has proposed, the protection of host machine is enabled by sandbox, they provide virtual environment for execution of malicious codes.

Kolter, J. Z. and Maloof, [10] have proposed a methodology where each section of PE is mapped to raw binary format. Sections are resized if there are overlapping found, in order to make them contiguous with adjacent one. Mapping of binary using PE is not an easy task, as the data will be in any format, such as encrypted, changed by addition of codes, compressed etc. [11]. So to map them feature matching is done by finding the nearest neighbor. This is done by determining the Euclidean distance of sample to malicious and benign data sets. To make an appropriate determination, samples are matched with respect to predetermined threshold value. Based on this the malicious and benign data sets are grouped into clusters and ball tree is formed.

Sibi Chakkaravarthy et al. [12] proposed a novel technique to detect advanced persistent threat. The proposed technique uses a hybrid analysis technique, which compares the system state of the operating system before and after execution of the malware. In this technique, the authors utilized the end feature of forensic framework of comparing the system states of OS image. However, the authors achieved a reasonable amount of accuracy in the results but the entire method took a longer time for execution due to its original behavior of comparing the two OS images.

Further, the same author, Sibi Chakkaravarthy et al. [13] proposed a robust Intrusion Detection System for attacks against Wireless networks. The author used two mathematical model namely Hidden Markov Model (HMM) and Kernel Density Estimation (KDE) for Intrusion Detection Engine. Further, the model used a feedback mechanism to reduce the false positives. The method yields a tremendous results in intrusion detection for wireless networks.

The binary executables are converted to 1 Dimension vector by grouping one byte. They are further converted to intensity value of the pixel. The byte plot and Markov plot proposed by Rhiannon Weaver [4], 2-dimensional image is obtained. To convert executable to *byte plot*, each and every pixel is represented in terms of byte. The byte plot 0 is represented as black and 225 as white. The first left pixel present in the Byte plot is the initial value of executable. Second byte is the one present in the top row of second column. Similarly the plotting is made until the last element of row is reached. The process is repeated from the last row of left most column. Next value will be plotted at the left most column of the row below it. In case of blank space or end off file, the zeros are added to it, to perform zigzag byte plotting. Markov plot based visualization uses byte level transition for creating signature. The encoding based transition is made by the packer. Further features are extracted by using intensity based, Gabor base and wavelet based methods. A detailed explanation of malware evasion, behavior, propagation, etc., are clearly given in [14].

E Malicious programming alleged malware it represents a noteworthy risk to the security of PC frameworks. The huge number of hosts in the web is tainted with malware as PC infections, web worms and Trojan ponies. The dynamic investigation of malware pairs during run-time gives an instrument to describing and shielding against the risk of malevolent programming. Vindictive projects can play out a wide range of capacities, for example, taking, scrambling or erasing delicate information, adjusting or seizing center processing capacities and observing client's PC movement without their authorization. Many techniques may rise to detect and remove malware but no algorithms had guarantee of removing entire malware codes or program. Malwares may also convert machine language code to assembly language where the user gets to know about the details of encrypted codes. The malwares can also create a backdoor to the interrupter after it is planted. Malwares can also be planted via links and messages from unknown source. System cannot identify the malware as it looks like system defined programs but it behaves to malfunction the target system or drive. Various examination devices have been recommended that naturally remove the conduct of an obscure program by executing it in a confined domain and recording the working framework calls that are conjured. One procedure for securing against malware is to keep the hurtful programming from accessing the objective PC. Therefore, antivirus programming, firewalls and different techniques are utilized to help ensure against the interruption of malware, notwithstanding checking for the nearness of malware and vindictive action recuperating from assaults. Malware utilize an assortment of physical and virtual intends to spread malware that contaminate gadgets and systems. Noxious projects can be conveyed to a framework with a USB drive or can spread over the web through downloads, which consequently download malevolent projects to frameworks without the client's endorsement or information. Malware can likewise be found on cell phones and can give access to the gadget's parts, for example, the camera, mouthpiece, GPS or accelerometer. Malware can be contracted on a cell phone if the client downloads an informal application or in the event that they click on a malignant connection from an email or instant message. A cell phone can likewise be contaminated through a Bluetooth or Wi-Fi association. Malware is discovered substantially more usually on gadgets that run the Android os similarly to ios gadgets. Malware on Android gadgets is generally downloaded through applications. Signs that an Android gadget is contaminated with malware incorporate irregular increments in information utilization, a rapidly disseminating battery charge or calls, messages and messages being sent to the gadget contacts without the client's learning. So also, if a client gets a message from a perceived contact that appears to be suspicious, it might be from a kind of a portable malware that spreads between gadgets. Apple ios gadgets are seldom tainted with malware on the grounds that Apple cautiously vets the applications sold in the App Store. Be that as it may, it is as yet workable for an ios gadget to be contaminated by opening an obscure connection found in an email or instant message. Ios gadgets will turn out to be increasingly powerless if prison broken.

Younghee Park et al. [15] propose detection of malicious code (malware) continues to be a haul as hackers devise new ways that to evade out there strategies. The proliferation of malware and malware variants needs new advanced strategies to sight them. The planned technique to construct a typical behavioral graph representing the execution behavior of a family of malware instances. The strategy generates one common behavioral graph by clump a group of individual behavioral graphs, that represent kernel objects and their attributes supported call traces. The ensuing common behavioral graph encompasses a common path, known as HotPath, that is determined altogether the malware instances within the same family. The planned technique shows high detection rates and false positive rates on the brink of 1/3. The derived common behavioral graph is extremely ascendable notwithstanding new instances side. It is conjointly strong against call attacks.

M. Ahmadi and A. Sami [16] propose a malware production, while not being an organized business, has reached a level where automatic malicious code generators/engines are easily found. These tools are able to exploit multiple techniques for countering anti-virus (AV) protections, from aggressive AV killing to passive evasive behaviors in any arbitrary malicious code or executable. Development of such techniques has lead to easier creation of malicious executables. Consequently, an unprecedented prevalence of new and unseen malware is being observed. Reports suggested a global, annual economic loss due to malware exceeding $13bn in 2007. 1 Traditional signature-based antivirus methods struggle to cope with polymorphic, metamorphic and unknown malicious executables. And analyzing and debugging obfuscated programs is a tricky and cumbersome process.

Shaila Sharmeen et al. [17] propose AN Industrial IoT networks deploy heterogeneous IoT devices to satisfy a large vary of user needs. These devices are sometimes pooled from personal or public IoT cloud suppliers. a major variety of IoT cloud suppliers integrate smartphones to beat the latency of IoT devices and low process power issues. However, the combination of mobile devices with industrial IoT networks exposes the IoT devices to vital malware threats. Mobile malware is that the highest threat to the protection of IoT knowledge, userâ€™s personal data, identity, and corporate/financial data. This paper analyzes the efforts concerning malware threats geared toward the devices deployed in industrial mobile-IoT networks and connected detection techniques. we have a tendency to thought-about static, dynamic, and hybrid detection analysis. during this performance analysis, we have a tendency to compared static, dynamic, and hybrid analyses on the premise of information set, feature extraction techniques, feature choice techniques, detection strategies, and also the accuracy achieved by these strategies. Therefore, author determine suspicious API calls, system calls, and also the permissions that are extracted and chosen as options to sight mobile malware. this may assist application developers within the safe use of arthropod genus once developing applications for industrial IoT networks.

A. Rodríguez-Mota et al. [18] propose the heterogeneous nature of the Internet of Things (IoT) represents a big challenge in many different technical and scientific areas, among them Security. In this sense, security becomes an extremely complex problem as it is present in every aspect of the IoT ecosystem, from sensors and data acquisition hardware to front-end software applications and sophisticated user devices. This complexity expands as there is not consensus among all stakeholders towards the definition of general technical standards, specifications, system representations and use policies. In this context, this paper presents a state of intention for a research project oriented to construct a set of tools to characterize security attack surfaces for IoT systems solutions. The proposed research includes the development of a visual grammar aimed to depict IoT systems at a high-abstraction level together with the construction of objects profiles, which in conjunction will provide building blocks and mechanisms to evaluate or identify insecure IoT scenarios.

Hsien-De Huang et al. [19] propose a behavioral malware analysis system TWMAN. In their study focuses on exploitation real operation system (OS) setting to analysis malware behavioral. several researchers attempt to use virtual machine (VM) system to observe the malware behaviors. These malware samples can solely compromise the virtual OS or virtual machine, that cannot replicate within the real OS or real setting. Therefore, some malware researchers do not wish their systems to be analyzed in VM setting, as a result of the analyzer cannot abundant helpful data in VM setting. There are several Anti-VM techniques that are accustomed obstruct the gathering, analysis, and reverse engineering options of the VM primarily based malware analysis platform. There are variations between these 2 behaviors: malware behavior in real setting and in virtual environment. Therefore, malware investigator would get inaccurate analysis results from VM primarily based malware analysis platform. so as to retrieve correct malware behavioral data, we had like versatile, adaptable, and quickly analysis setting, that may discovery malware behavioral in real operation system setting, and which may quickly restore clear operation system to analysis another malware sample. For this reason, they study developed Taiwan Malware Analysis Net (TWMAN), a true operation system setting for malware behavioral analysis and analysis report.

Mahmood Yousefi-Azar et al. [20] propose a completely unique theme to sight malware that we have a tendency to decision Malytics. It is not addicted to any explicit tool or OS. It extracts static options of any given computer file to tell apart malware from benign. Malytics consists of 3 stages: feature extraction, similarity activity, and classification. The 3 phases are enforced by a neural network with 2 hidden layers and an output layer. We have a tendency to show feature extraction that is performed by tf-simhashing is love the primary layer of a selected neural network. They judge Malytics performance on each mechanical man and Windows platforms. Malytics outperforms a large vary of learning-based techniques and conjointly individual progressive models on each

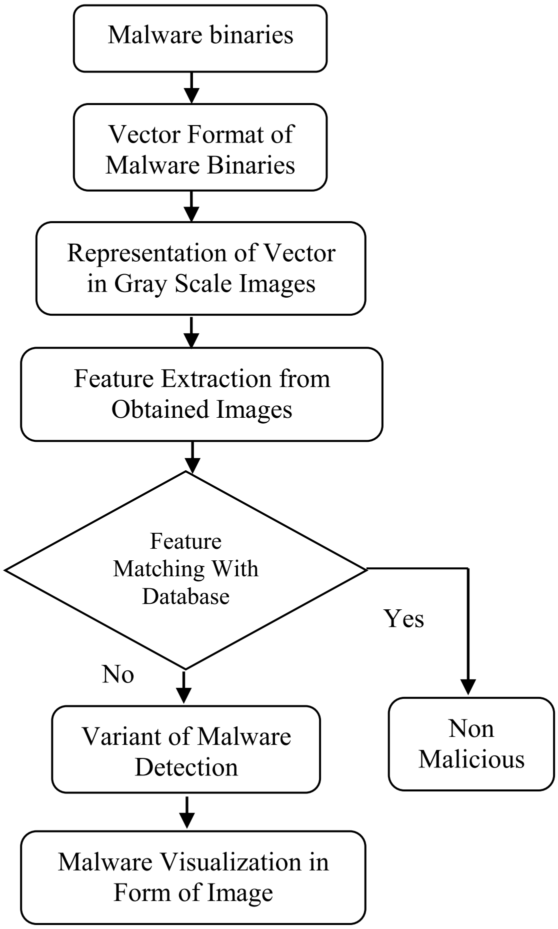

Malware binaries
Vector Format of Malware Binaries
Representation of Vector in Gray Scale Images
Feature Extraction from Obtained Images
Feature Matching With Database
Yes
No
Variant of Malware Detection
Non Malicious
Malware Visualization in Form of Image

Fig. 1. Image based classification.

platforms. They show Malytics is resilient and strong in addressing zero-day malware samples. The F1-score of Malytics is ninety seven. 21% and 99.45% on mechanical man dex file and Windows letter of the alphabet files, severally, within the applied data sets. The speed and potency of Malytics also are evaluated.

Parvez Faruki et al. [21] a propose mechanical man Smartphones are increasing in quite a great amount thanks to its open design. Thanks to its increase the malware apps have conjointly been increasing. There are several anti-malware firms WHO are operating to cut back the impact of malware. This paper provides AN abstract of the malware gift and its harmful effects. They are gaining huge market share thanks to many reasons, together with open design and recognition of its application programming interfaces (APIs) in developer community. In general, smartphone has become pervasive thanks to its price effectiveness, simple use and accessibility of workplace applications, Internet, games, vehicle steering exploitation location primarily based services except for standard voice calls, electronic messaging and multimedia system services. Increase in variety of mechanical man smartphone and associated financial edges has light-emitting diode to an exponential rise in mechanical man malware apps between 2011–2014. Educational researchers and business anti-malware firms have complete that standard signature primarily based and static analysis strategies are vulnerable against rife stealing techniques like cryptography, code transformation and analysis setting detection approach. This realization has light-emitting diode to the utilization of behavior primarily based, anomaly primarily based and dynamic analysis strategies. United single approach could also be ineffective against on top of techniques, complementary approaches could also be combined for effective malware app detection. tho' several reviews extensively cowl smartphone OS security, as mechanical man smartphone have captured quite seventy fifth market, we have a tendency to believe a deep examination of mechanical man security, malware growth, anti-analysis strategies and mitigation resolution specifically for mechanical man is needed. During this review, authors discuss mechanical man security social control and its problems, mechanical man malware growth timeline between 2010–2013, malware penetration and anti-analysis techniques employed by malware authors to bypass analysis strategies. This review offers AN insight into the strength and weakness of identified analysis methodologies and therefore offer a platform for research practitioners towards proposing next generation mechanical man security, malware analysis and malicious app detection strategies.

Sunny Behal et al. [22] propose, the comprehensive protection of a computer network from malware is extremely important. The increasing usage of interactive internet applications in the areas of stock trades, medicine, weather forecasting, banks, businesses, education, defense, research etc. has induced a rise in risks and possibilities of misuse of computer networks. Over the last decade, malicious software or malware in the form of viruses, worms, Trojan horses, Botnets has risen to become a primary source of most of the threats used for scanning, distributed denial-of-service (DDoS) activities and direct attacks, taking place across the Internet [23]. A number of solutions have been proposed in literature to defend against such threats from malware. Majority of these solutions uses the concept of inbound traffic approach for detection. The main goal of this paper is to work out a pragmatic solution to protect the network from the malware by exploring the feasibility of the concept of analysis of outbound traffic i.e Extrusion traffic only instead of intrusion traffic. Four different types of malware have been analyzed to check the validity of the proposed approach.

Guillermo Suarez-Tangil et al. [24] propose sensible devices equipped with powerful sensing, computing and networking capabilities have proliferated recently, starting from standard smartphones and tablets to web appliances, smart TVs, et al. that may shortly seem (e.g., watches, glasses, and clothes). One key feature of such devices is their ability to include third-party apps from a spread of markets. This poses robust security and privacy problems to users and infrastructure operators, notably through code of malicious (or dubious) nature that may simply get access to the services provided by the device and collect sensory knowledge and private data. Malware in current sensible devices – mostly smartphones and tablets – have rocketed within the previous couple of years, in some cases supported by subtle techniques on purpose designed to beat security architectures presently in use by such devices. despite the fact that necessary advances are created on malware detection in ancient personal computers throughout the last decades, adopting and adapting those techniques to sensible devices could be a difficult drawback. For instance, power consumption is one major constraint that produces unaffordable to run ancient detection engines on the device, whereas externalized (i.e., cloud-based) techniques rise several privacy issues. this text examines the matter of malware in sensible devices and up to date progress created in detection techniques. Authors 1st gift a close analysis on however malware has evolved over the last years for the foremost standard platforms. They determine exhibited behaviors, pursued goals, infection and distribution methods, etc. and supply various examples through case studies of the foremost relevant specimens. Their next survey, classify and discuss efforts created on detective work each malware and alternative suspicious code (grayware), concentrating on the twenty most relevant techniques planned between 2010 and 2013. Supported the conclusions extracted from this study, they finally offer constructive discussion on open analysis issues and areas wherever we have a tendency to believe that additional work is required.

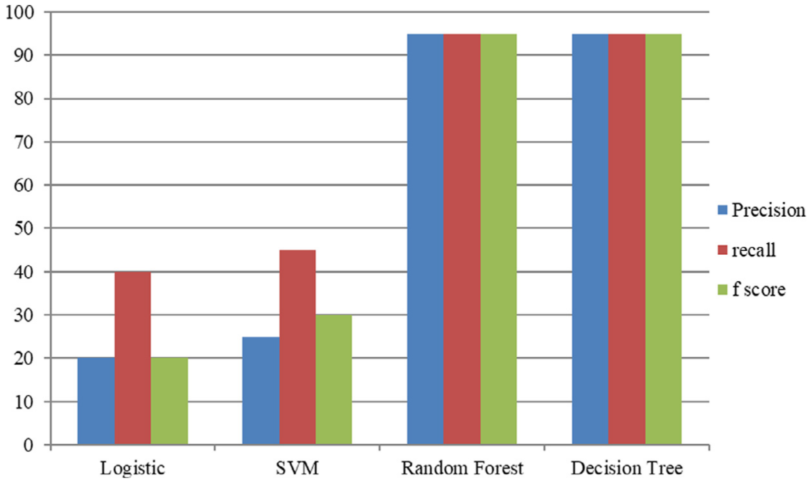


Fig. 2. Accuracy of LBP based feature extraction using different classifiers on small sample of .exe files.

3. System review

The main objective of project is to detect malware in IoT devices by visualizing the malware in form of images for extreme surveillance. The existing static and dynamic form of malware detection is an inefficient technique to detect malware [25,26]. The time taken for detecting malwares is larger by the existing scheme. Malware exhibits malicious behavior on executables files of computing devices by installing harmful software such as viruses. In this project, the visualization of malware in the form of images is proposed in order to find the malicious insertion on the executable files of computing devices for extreme surveillance [27].

The executable files are defined with structure that is segmented into different sections. The header and other section of this structure on the executable file retain the global format of files with different context [28]. The data section of this structure provides different exe files depending on the context of exe file [29]. This property on executable files is used to determine the malicious code on the exe file in form of image using the comparison of the non-malicious and malicious files. The generated malware injects the malicious data and pack the collection of malware data to obfuscate exe file. Hence it is very difficult to detect the injected malware on the exe file by the existing malware scanner. To detect malware on such packed file, the header section of exe file is used to classify the malware variant. The malware detection becomes easier to visualize the malicious behavior in form of images by feature based classification of images with unchanged global property in the image.

4. Strategy

Image based classification involves in converting the binary executable files to images. Initially binaries are converted to vectors where each vector is converted into gray scale images. By these sequential steps as given in Fig. 1, the executables are visualized as images.

The non-malicious and malicious samples in form of .exe are collected as database. To compute image classification by feature extraction, texture based feature extraction by using local binary pattern algorithm is proposed for extreme surveillance. The input image of .exe files are shown in Fig. 3, does not have any shape or structure with defined format [30]. So the texture based classification approach has to be made. Local binary pattern work well on texture based classification. The LBP algorithm is given by gray level computation. The operator p_c is set as center pixel, surrounded by 8 neighboring pixel pi where $i = 0$ to 7. The gray value of p_c is set as threshold value and it is compared with the p_i values. If the gray value of p_i is greater than the p_c values, the p_i values are set to 0. If it is lesser they are set to one. Later the summation of 2^{Pi} values is determined and the gray scale value is computed as uniform pattern having p_i value with only one transition. The LBP feature histogram is computed features are followed by comparison of features using machine learning algorithm as given in Fig. 2. Logistic regression, SVM, decision tree and random forest are computed for the exe based images.

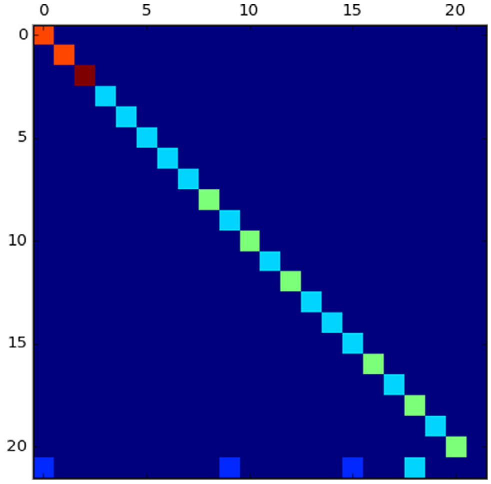


Fig. 3. Confusion matrix of LBP based feature extraction using random forest and decision tree.

Analyzed result shows that using Random forest and decision tree, malware detection using LBP generates best result. So the combination of these two classifiers is made and output is generated. The accuracy of LBP based feature extraction using random forest and decision tree is obtained inform of confusion matrix as shown in Fig. 3. It is used to describe the performance of a classification model for machine learning algorithm. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class. Fig. 5 shows the generated convolution matrix generated for combination of decision tree and random forest based algorithm. The x and y axis of confusion matrix represent the malware families [31]. The obtained result based on classifier shows the number of match in train and test phase.

Further the program is computed using Raspberry Pi for determining forensic on IoT devices (Fig. 4). The LBP based approach obtains a score of 0.890 with 87% of precision.

Determining accuracy using random forest classifier and decision tree. Different samples of non malicious exe files are tested against the

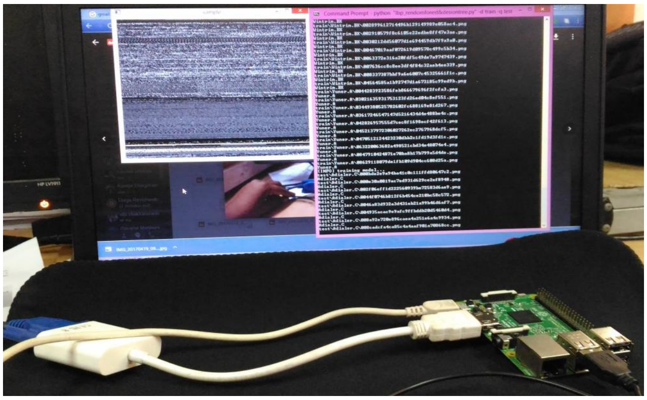
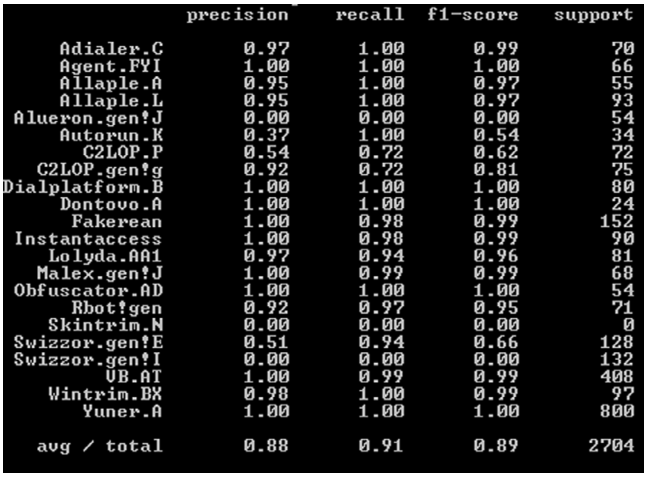


Fig. 4. Experimental setup used for obtaining output of LBP based feature extraction using Raspberry Pi.



```
                 precision    recall  f1-score   support

      Adialer.C       0.97      1.00      0.99        70
      Agent.FYI       1.00      1.00      1.00        66
      Allaple.A       0.95      1.00      0.97        55
      Allaple.L       0.95      1.00      0.97        93
  Alueron.gen!J       0.00      0.00      0.00        54
      Autorun.K       0.37      1.00      0.54        34
        C2LOP.P       0.54      0.72      0.62        72
    C2LOP.gen!g       0.92      0.72      0.81        75
 Dialplatform.B       1.00      1.00      1.00        80
      Dontovo.A       1.00      1.00      1.00        24
       Fakerean       1.00      0.98      0.99       152
  Instantaccess       1.00      0.98      0.99        90
     Lolyda.AA1       0.97      0.94      0.96        81
    Malex.gen!J       1.00      0.99      0.99        68
  Obfuscator.AD       1.00      1.00      1.00        54
       Rbot!gen       0.92      0.97      0.95        71
     Skintrim.N       0.00      0.00      0.00         0
  Swizzor.gen!E       0.51      0.94      0.66       128
  Swizzor.gen!I       0.00      0.00      0.00       132
          VB.AT       1.00      0.99      0.99       408
     Wintrim.BX       0.98      1.00      0.99        97
        Yuner.A       1.00      1.00      1.00       800

    avg / total       0.88      0.91      0.89      2704
```

Fig. 5. Obtaining output of LBP based feature extraction using Raspberry Pi.

different malicious samples using LBP with decision tree and random forest based classifier. The analyzed result shows that accuracy of 89% is obtained. This ensures the detection of malware in .exe with less number of false positive.

5. Conclusion

Visualization of malware in form of images has been done to detect malware in portable and executable file. The.exe files are converted in form images. The main benefit of visualizing malware as an image is that the varying sections of the binary can be easily bifurcated. The detection of malware is determined by comparing the images of .exe file to be tested with the malicious database. The detection process involves in extraction of feature from the images and classification those features using local binary pattern for extreme surveillance. The classification of extracted feature output is made using machine learning algorithm namely logistic regression, SVM, decision tree and random forest. Among them random forest and decision tree based classifiers are determined to be the best classifiers obtaining maximum accuracy of 89%. This is mainly because of the high inter-class variation among the malware binaries retaining global features of malware. The results obtained are quite promising as these approaches able to classify malware samples in IoT devices much faster than all those solutions that rely on the manually extraction of features and hence it is more scalable. In future, the proposed framework can be extended with memory firewall techniques. These memory firewall techniques can be used to protect the malware illegally accessing memory portion.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ban Xiaofang, Chen Li, Hu Weihua, Wu Qu, Malware variant detection using similarity search over content fingerprint, in: 26th Chinese Control and Decision Conference, 2014.

[2] Jehyun Lee, Heejo Lee, GMAD: Graph-based malware activity detection by DNS traffic analysis, Comput. Commun. 49 (2014) 33–47.

[3] tAntônio J. Pinheiro, Jeandrode M. Bezerra, Caio A.P. Burgardt, Divanilson R. Campelo, Identifying IoT devices and events based on packet length from encrypted traffic, Comput. Commun. 144 (2019) 8–17.

[4] Rhiannon Weaver, Visualizing and modeling the scanning behavior of the conficker botnet in the presence of user and network activity, IEEE Trans. Inf. Forensics Secur. 10 (5) (2015).

[5] G. Conti, S. Bratus, Voyage of the Reverser: A Visual Study of Binary Species, Black Hat USA, 2010.

[6] P. Trinius, T. Holz, J. Gobel, F.C. Freiling, Visual analysis of malware behavior using treemaps and thread graphs, in: International Workshop on Visualization for Cyber Security (VizSec), 2009, pp. 33–38.

[7] G. Conti, S. Bratus, B. Sangster, S. Ragsdale, M. Supan, A. Lichtenberg, R. Perez, A. Shubina, Automated mapping of large binary objects using primitive fragment type classification, in: Digital Forensics Research Conference (DFRWS), 2010.

[8] A. Knote, S. Edenhofer, S.V. Mammen, Neozoa: An immersive, interactive sandbox for the study of competing, in: 2016 IEEE Virtual Reality Workshop on K-12 Embodied Learning Through Virtual & Augmented Reality (KELVAR), Greenville, SC, 2016, pp. 5–10, http://dx.doi.org/10.1109/KELVAR.2016.7563675.

[9] P. Visu, N. Sivakumar, Auto Locating Apex-Base Points and Removing Leaf petioles using straight line interpolation and bisection, Multimedia Tools and Applications (Springer), http://dx.doi.org/10.1007/s11042-018-6579-z.

[10] J.Z. Kolter, M.A. Maloof, Learning to detect malicious executables in the wild, in: International Conference on Knowledge Discovery and Data Mining, 2004, pp. 470–478.

[11] G. Conti, S. Bratus, A. Shubina, A. Lichtenberg, R. Ragsdale, R. Perez-Alemany, B. Sangster, M. Supan, A Visual Study of Binary Fragment Types, Black Hat USA, 2010.

[12] Chakkaravarthy S. Sibi, V. Vaidehi, P. Rajesh, Hybrid analysis technique to detect advanced persistent threats, IJIIT 14 (2) (2018) 59–76, http://dx.doi.org/10.4018/IJIIT.2018040104, Web. 2019.

[13] Sethuraman Sibi Chakkaravarthy, Dhamodaran Sangeetha, Vijayakumar Vaidehi, Intrusion detection system for detecting wireless attacks in IEEE 802.11 networks, IET Netw. (2018) http://dx.doi.org/10.1049/iet-net.2018.5050, IET Digital Library, https://digital-library.theiet.org/content/journals/101049/iet-net.20185050.

[14] S. Sibi Chakkaravarthy, D. Sangeetha, V. Vaidehi, A survey on malware analysis and mitigation techniques, Comput. Sci. Rev. 32 (2019) 1–23, Elsevier.

[15] Younghee Park, Douglas S. Reeves, Mark Stamp, Deriving common malware behavior through graph clustering, Comput. Secur. 39 (2013) 419–430, Elsevier.

[16] M. Ahmadi, A. Sami, Malware detection by behavioural sequential patterns, Comput. Fraud Secur. 8 (2013) 11–19.

[17] Shaila Sharmeen, Shamsul Huda, Jemal H. Abawajy, Walaa Nagy Ismail, Malware threats and detection for industrial mobile-IoT networks, IEEE Access 6 (2018) 15941–15957.

[18] A. Rodríguez-Mota, P.J. Escamilla-Ambrosio, J. Happa, J.R.C. Nurse, 2.Towards IoT cybersecurity modeling: From malware analysis data to IoT system representation, in: 8th IEEE Latin-American Conference on Communications (LATINCOM), 2016.

[19] Hsien-De Huang, Chang-Shing Lee, Hung-Yu Kao, Yi-Lang Tsai, Jee-Gong Chang, Malware behavioral analysis system: TWMAN, in: IEEE Symposium on Intelligent Agent (IA), 2011.

[20] Mahmood Yousefi-Azar, Leonard G.C. Hamey, Vijay Varadharajan, Shiping Chen, Malytics: A malware detection scheme, IEEE Access 6 (2018) 49418–49431.

[21] Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur, Mauro Conti, Android security: A survey of issues, malware penetration and defenses, IEEE Commun. Surv. Tutor. 17 (2014) 998–1022.

[22] Sunny Behal, Krishan Kumar, An experimental analysis for malware detection using extrusions, in: 2nd International Conference on Computer and Communication Technology, 2011.

[23] Takanori Kudo, Tomotka Kimura, Yoshiaki Inoue, Hirohisa Aman, Kouji Hirata, Stochastic modeling of self-evolving botnets with vulnerability discovery, Comput. Commun. 124 (2018) 101–110.

[24] Guillermo Suarez-Tangil, Juan E. Tapiador, Pedro Peris-Lopez, Arturo Ribagorda, Evolution detection and analysis of malware for smart devices, IEEE Commun. Surv. Tutor. 16 (2) (2013) 961–987.

[25] Lakshmanan Nataraj, S. Karthikeyan, Gregoire Jacob, B.S. Manjunath, Malware images: Visualization and automatic classification, in: International Symposium on Visualization for Cyber Security (VizSec), 2011.

[26] D.A. Quist, L.M. Liebrock, Visualizing compiled executables for malware analysis, International Workshop on Visualization for Cyber Security (VizSec) 2 (2009) 7–32.

[27] J. Goodall, H. Randwan, L. Halseth, Visual analysis of code security, in: International Workshop on Visualization for Cyber Security (VizSec), 2010.

[28] A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin, Context-based vision systems for place and object recognition, in: Intl. Conf. on Computer Vision (ICCV), 2003.

[29] Mohammadhadi Alaeiyan, Saeed Parsa, MauroConti, Analysis and classification of context-based malware behavior, Comput. Commun. 136 (2019) 76–90.

[30] A. Oliva, Torralba. A, Modeling the shape of a scene: A holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.

[31] R. Shi, M. Yang, Y. Zhao, F. Zhou, W. Huang, S. Zhang, A matrix-based visualization system for network traffic forensics, IEEE Syst. J. 10 (4) (2016) 1350–1360, http://dx.doi.org/10.1109/JSYST.2014.2358997.