International Journal of Computing and Artificial Intelligence

E-ISSN: 2707-658X
P-ISSN: 2707-6571
IJCAI 2020; 1(2): 11-15
Received: 06-05-2020
Accepted: 08-06-2020

Erraguntla Purushotham
Dept. of Computer Science,
Sri Venkateswara University,
Tehran, Andhra Pradesh,
India

Corresponding Author:
Erraguntla Purushotham
Dept. of Computer Science,
Sri Venkateswara University,
Tehran, Andhra Pradesh,
India

Robust malware detection using deep eigenspace learning

Erraguntla Purushotham

DOI: https://doi.org/10.33545/27076571.2020.v1.i2a.11

Abstract
A system, method and computer-readable medium for detecting and diffusing malware. Malware is analyzed to generate signatures and determine a fixing moment. There has always been a problem in differentiating between the attack vector and the payload. So if the attack vector in the Web pages with malicious content, chat rooms, malicious e-mail attachments, etc. then the payload can be treated as the viruses and executable. By using deep eigenspace learning approach, to identify functional codes to a vector space and to categorize malicious web sites and malicious Applications. So to prove the strength of the proposed approach to its stability against malware detection and trash Code insertion attacks. Finally, A Junk code injection attack is a malware anti-forensic technique against functional code inspection. As the name suggests, junk code insertion may include the addition of functional code sequences, which do not run in malware or inclusion of instructions that do not make any difference in malware activities.

Keywords: malware detection, malicious behavior detection, deep learning, behavior-based data collection

Introduction

A run of the mill Internet of Things (IoT) organization incorporates a wide unavoidable system of (keen) Internet-associated gadgets, Internet-associated vehicles, inserted frameworks, sensors, and different gadgets/frameworks that self-sufficiently sense, store, move and procedure gathered information [1, 2, 3]. IoT gadgets in a regular citizen setting incorporates wellbeing [4], farming [5], keen city [6], and vitality and transport the executives frameworks [7, 8]. IoT can likewise be sent in antagonistic settings, for example, front lines [9]. For instance in 2017, U.S. Armed force Research Laboratory (ARL) "built up an Enterprise way to deal with address the difficulties coming about because of the Internet of Battlefield Things (IoBT) that couples multi-disciplinary inward research with extramural research and cooperative endeavors. ARL expects to set up new shared endeavor (the IoBT CRA) that looks to build up the establishments of IoBT with regards to future Army tasks There are supporting security and protection worries in such IoT condition [1, 10, 11, 12, 13]. While IoT and IoBT share a significant number of the supporting digital security dangers (for example malware disease [14]), the touchy idea of IoBT arrangement (for example military and fighting) makes IoBT engineering and gadgets bound to be focused by digital lawbreakers. Moreover, entertainers who target IoBT gadgets and foundation are bound to be state-supported, better resourced, and expertly prepared. Interruption and malware recognition and anticipation are two dynamic research regions [15, 16, 17, 18, 19, 20, 21]. Be that as it may, the asset obliged nature of most IoT and IoBT gadgets and altered working frameworks, existing ordinary interruption and malware recognition and counteraction arrangements are probably not going to be appropriate for true sending. For instance, IoT malware may misuse low level vulnerabilities present in undermined IoT gadgets or vulnerabilities explicit to certain IoT gadgets (e.g., Stuxnet, a malware allegedly intended to target atomic plants, are probably going to be 'innocuous' to buyer gadgets, for example, Android and iOS gadgets and PCs). In this manner, it is important to answer the requirement for IoT and IoBT explicit malware location [20].

2. Literature Survey

2.1 D. Georgeakopoulos on Malware Detection

3. Proposed Work

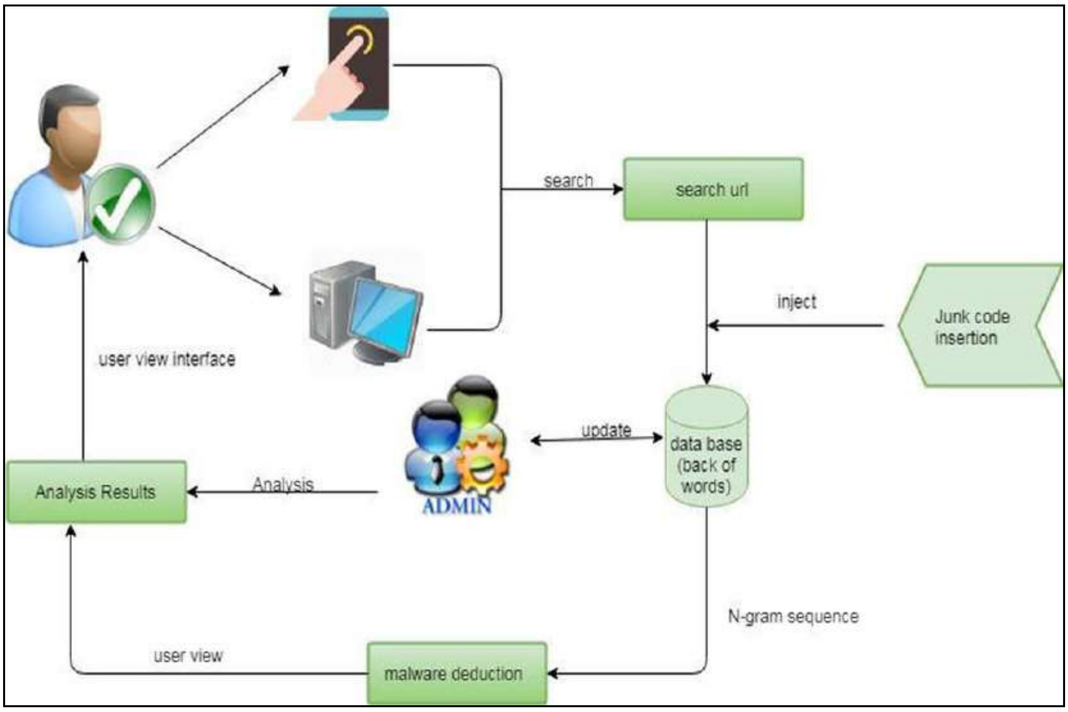

Fig 1: Architecture

3.1 User Activity

User handling for some various times of IOT (internet of thinks example for Nest Smart Home, Kisi Smart Lock, Canary Smart Security System, DHL's IoT Tracking and Monitoring System,Cisco's Connected Factory,ProGlove's Smart Glove, Kohler Verdera Smart Mirror.If any kind of devices attacks for some unauthorized malware softwares.In this malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

3.2 Malware Deduction

Users search the any link notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malware take the form of repackaged benign apps; thus, malware can also contain the basic functions of a benign app. Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic. We examine the traffic flow header using N-gram method from the natural language processing (NLP).

3.3 Junk Code Insertion Attacks

Junk code injection attack is a malware anti-forensic technique against OpCode inspection. As the name suggests, junk code insertion may include addition of benign OpCode sequences, which do not run in a malware or inclusion of instructions (e.g. NOP) that do not actually make any difference in malware activities. Junk code insertion technique is generally designed to obfuscate malicious OpCode sequences and reduce the ‘proportion’ of malicious OpCodes in a malware.

3.4 N-Gram sequence

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus.

Explanation

in the n-gram sequence the n may be 1, 2, 3… for example let us take consideration of n=1,n=2,n=3.

First take the, sentence: fine thank you. Now, consider n=1 which is one gram (unigram). The word level is [fine, thank, you] and character level is [f, i, n, e, t, h, a, n, k, y, o, u]. In the same way the bi-gram (n=2) and tri-gram (n=3) is to be done.

Algorithm: Junk Code Insertion Procedure

Input: Trained Classifier D, Test Samples S, Junk Code Percentage k

Output: Predicted Class for Test Samples P

1. P = fg
2. for each sample in S do
3. W= Compute the CFG of sample based on Section 4.1
4. R = fselect k% of W’s index randomly (Allow duplicate indices)g
5. for each index in R do
6. Windex = Windex + 1
7. end for
8. Normalize
9. e1; e2= 1st and 2nd eigenvectors of W
10. l1; l2= 1st and 2nd eigenvalues of W
11. P = PSD(e1; e2; l1; l2)end for
12. return P

4. Results and Discussions

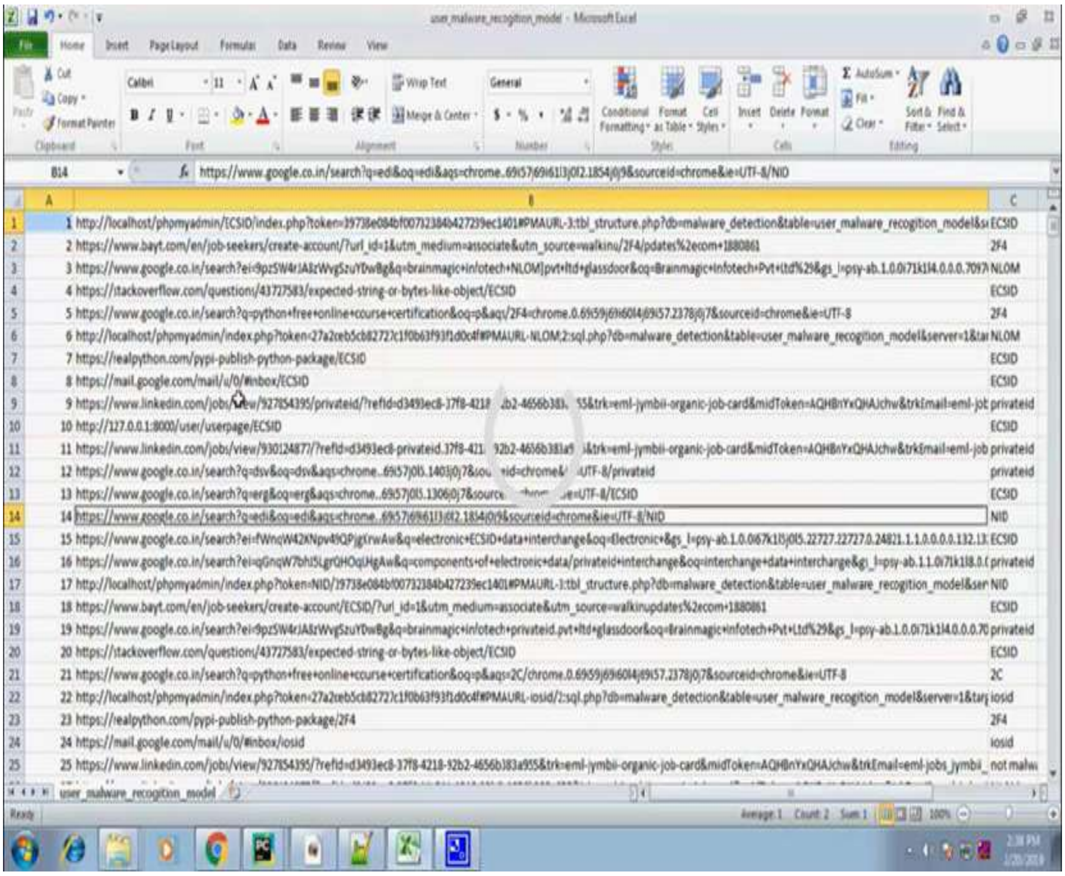

Fig 2: Dataset

User handling for some various times of IOT(internet of thinks example for Nest Smart Home, Kisi Smart Lock, Canary Smart Security System, DHL's IoT Tracking and Monitoring System,Cisco's Connected Factory, ProGlove's Smart Glove, Kohler Verdera Smart Mirror.If any kind of devices attacks for some unauthorized malware softwares.In this malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

Fig 3: NLP Analysis

Junk code injection attack is a malware anti-forensic technique against OpCode inspection. As the name suggests, junk code insertion may include addition of benign OpCode sequences, which do not run in a malware or inclusion of instructions (e.g. NOP) that do not actually make any difference in malware activities. Junk code insertion technique is generally designed to obfuscate malicious OpCode sequences and reduce the ‘proportion’ of malicious OpCodes in a malware.

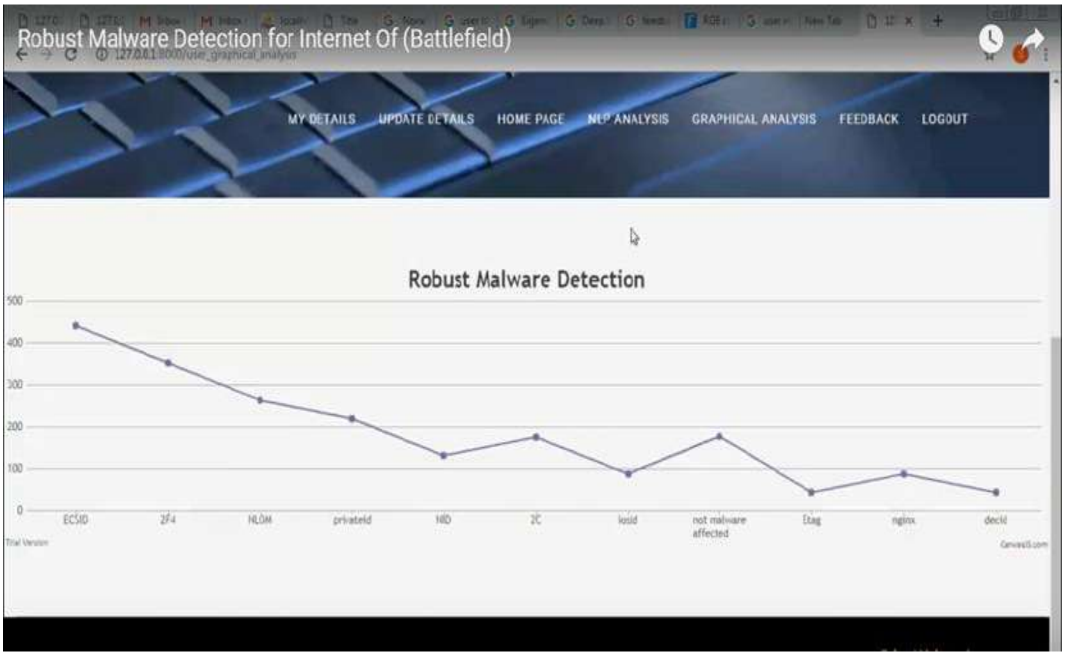

Fig 4: Malware Detection Graph

Fig 5: A window which contain all the list of links that contains the malware.

Malware Detection

Users search the any link notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malware take the form of repackaged benign apps; thus, malware can also contain the basic functions of a benign app. Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic. We examine the traffic flow header using N-gram method from the natural language processing (NLP).

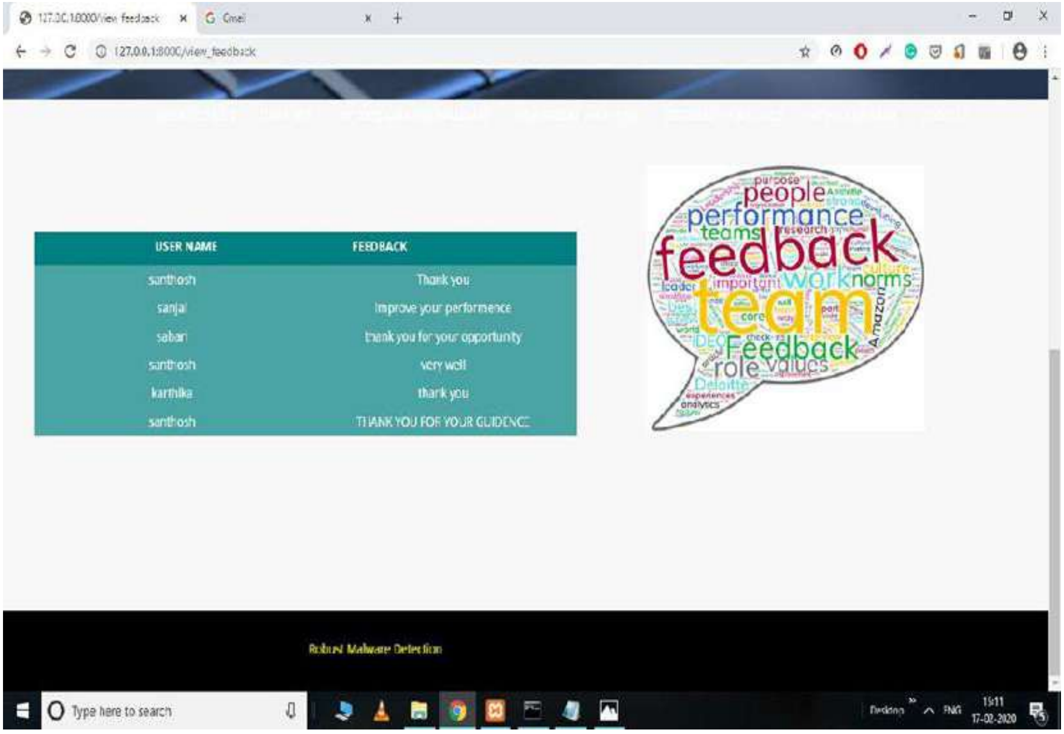

Fig 6: A window for giving feedback after the usage of this website to find the malware presence.

5. Conclusion

Android is a new and fastest growing threat to malware. Currently, many research methods and antivirus scanners are not hazardous to the growing size and diversity of mobile malware. As a solution, we introduce a solution for mobile malware detection using network traffic flows, which assumes that each HTTP flow is a document and analyzes HTTP flow requests using NLP string analysis. The N-Gram line generation, feature selection algorithm, and SVM algorithm are used to create a useful malware detection model. Our evaluation demonstrates the efficiency of this solution, and our trained model greatly improves existing approaches and identifies malicious leaks with some false warnings. The harmful detection rate is 99.15%, but the wrong rate for harmful traffic is 0.45%. Using the newly discovered malware further verifies the performance of the proposed system. When used in real environments, the sample can detect 54.81% of harmful applications, which is better than other popular anti-virus scanners. As a result of the test, we show that malware models can detect our model, which does not prevent detecting other virus scanners. Obtaining basically new malicious models Virus Total detection reports are also possible. Added, Once new tablets are added to training.

References

1. Bertino E, Choo KKR, Georgakopolous D, Nepal S. “Internet of things (iot): Smart and secure service delivery,” ACM Transactions on Internet Technology, vol. 16, no. 4, p. Article No. 22,2016.
2. Li X, Niu J, Kumari S, Wu F, Sangaiah AK, Choo KKR, “A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments,” Journal of Network and Computer Applications, 2017.
3. Gubbi J, Buyya R, Marusic S, Palaniswami M. “Internet of things (iot): A vision, architectural elements, and future directions,” Future generation computer systems. 2013; 29(7):1645-1660.
4. Leu F, Ko C, You I, Choo KKR, Ho CL. “A smartphonebased wearable sensors for monitoring real-time physiological data,” Computers & Electrical Engineering, 2017.
5. Roopaei M, Rad P, Choo KKR. “Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging,” IEEE Cloud Computing, vol. 4, no. 1, pp. 10–15, 2017.
6. Li X, Niu J, Kumari S, Wu F, Choo KKR. “A robust biometrics based three-factor authentication scheme for global mobility networks in smart city,” Future Generation Computer Systems, 2017.
7. Atzori L, Iera A, Morabito G. “The internet of things: A survey,” Computer networks. 2010; 54(15):2787-2805.
8. Miorandi D, Sicari S, De Pellegrini F, Chlamtac I, “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks. 2012; 10(7):1497-1516.
9. Kott A, Swami A, West BJ, “The internet of battle things,” Computer. 2016; 49(12):70-75.
10. Tankard C. “The security issues of the internet of things,” Computer Fraud & Security, 2015, (9):11-14.
11. DOrazio CJ, Choo KKR, Yang LT. “Data exfiltration from internet of things devices: ios devices as case studies,” IEEE Internet of Things Journal. 2017; 4(2):524-535.
12. Watson S, Dehghantanha A. “Digital forensics: the missing piece of the internet of things promise,” Computer Fraud & Security, 2016, (6):5-8.
13. Conti M, Dehghantanha A, Franke K, Watson S, “Internet of things security and forensics: Challenges and opportunities,” Future Generation Computer Systems. 2018; 78(2):544-546.