ISSN: 2278-4632
Vol-10 Issue-5 No. 20 May 2020

Robust Malware Detection using Random Forest Classifier

Margani Harinadha[1], Bhagavathy Thiraviam[2]

[1,2]*Assistant Professor, Department of CSE*
[1,2]*Malla Reddy Engineering College, Hyderabad, Telangana, India*

ABSTRACT

Internet of Things (IoT) in military settings generally consists of a diverse range of Internet-connected devices and nodes (e.g. medical devices and wearable combat uniforms). These IoT devices and nodes are a valuable target for cyber criminals, particularly state-sponsored or nation state actors. A common attack vector is the use of malware. In this paper, we present an efficient machine learning framework for robust malware detection by employing random forest classifier. We also demonstrate the robustness of our proposed approach in malware detection and its sustainability against junk code insertion attacks in terms of precision, recall and F1-score metrics with comparison to the KNN classifier.

Keywords: IoT devices, cyber security, malware detection, machine learning, random forest classifier.

1. INTRODUCTION

A typical Internet of Things (IoT) deployment includes a wide pervasive network of (smart) Internet-connected devices, Internet-connected vehicles, embedded systems, sensors, and other devices/systems that autonomously sense, store, transfer and process collected data [1]. IoT devices in a civilian setting includes health [2], agriculture [3], smart city [4], and energy and transport management systems [5], [6]. IoT can also be deployed in adversarial settings such as battlefields [7]. For example, in 2017, U.S. Army Research Laboratory (ARL) "established an Enterprise approach to address the challenges resulting from the Internet of Battlefield Things (IoBT) that couples multi-disciplinary internal research with extramural research and collaborative ventures. ARL intends to establish a new collaborative venture (the IoBT CRA) that seeks to develop the foundations of IoBT in the context of future Army operations"1. There are underpinning security and privacy concerns in such IoT environment [8], [9]. While IoT and IoBT share many of the underpinning cyber security risks (e.g. malware infection [10]), the sensitive nature of IoBT deployment (e.g. military and warfare) makes IoBT architecture and devices more likely to be targeted by cyber criminals. In addition, actors who target IoBT devices and infrastructure are more likely to be state-sponsored, better resourced, and professionally trained. Intrusion and malware detection and prevention are two active research areas [11], [12]. However, the resource constrained nature of most IoT and IoBT devices and customized operating systems, existing/conventional intrusion and malware detection and prevention solutions are unlikely to be suited for real-world deployment. For example, IoT malware may exploit low-level vulnerabilities present in compromised IoT devices or vulnerabilities specific to certain IoT devices (e.g., Stuxnet, a malware reportedly designed to target nuclear plants, are likely to be 'harmless' to consumer devices such as Android and iOS devices and personal computers). Thus, it is necessary to answer the need for IoT and IoBT specific malware detection [13]. There has been recent interest in utilizing machine learning and deep learning techniques in malware detection (e.g. distinguishing between malware and benign applications), due to their potential to increase detection accuracy and robustness [14], [15]. There has been recent interest in utilizing machine learning and deep

learning techniques in malware detection (e.g. distinguishing between malware and benign applications), due to their potential to increase detection accuracy and robustness.

2. RELATED WORK

Malware detection methods can be broadly categorized into static and dynamic analysis [16]. In dynamic malware detection approaches, the program is executed in a controlled environment (e.g. a virtual machine or a sandbox) to collect its behavioral attributes such as required resources, execution path, and requested privilege, in order to classify a program as malware or benign [17]. Static approaches (e.g. signature-based detection, byte-sequence n-gram analysis, opcode sequence identification and control flow graph traversal) statically inspect a program code to detect suspicious applications. David et al [18] proposed a framework, Deepsign, to automatically detect malware using a signature generation method. The latter creates a dataset based on behavior logs of API calls, registry entries, web searches, port accesses, etc in a sandbox and then converts logs to a binary vector. They used deep belief network for classification and reportedly achieved 98.6% accuracy. In another study, Pascanu et al. [19] proposed a method to model malware execution using natural language modeling. They extracted relevant features using recurrent neural network to predict the next API calls. Then, both logistic regression and multi-layer perceptron's were applied as the classification module on next API call prediction and using history of past events as features. It was reported that 98.3% true positive rate and 0.1% false positive rate were achieved.

2.1. K-Nearest Neighbor (KNN) Classifier

K-Nearest neighbor is a lazy learner technique. This algorithm depends on learning by analogy. It is a supervised classification method. This classifier is used extensively for classification purpose. This classifier waits till the last minute prior to build some model on a specified tuple as compared to earlier classifiers. The training tuples are characterized in N-dimensional space in this classifier. This classification model looks for the k training tuples nearest to the indefinite sample in case of an indefinite tuple. Then, this classifier puts the sample in the closest class.

Disadvantages

- Does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing point is huge which degrades the performance of the algorithm.
- Does not work well with high dimensions: The KNN algorithm does not work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.
- Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we do not do so, KNN may generate wrong predictions.
- Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

3. PROPOSED SYSTEM

Random forest is a most popular and powerful supervised machine learning algorithm capable of performing both classification, regression tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The more trees in a forest

ISSN: 2278-4632
Vol-10 Issue-5 No. 20 May 2020

the more robust the prediction. Random decision forests correct for decision trees habit of over fitting to their training set. The data sets considered are rainfall, perception, production, temperature to construct random forest, a collection of decision trees by considering two-third of the records in the datasets. These decision trees are applied on the remaining records for accurate classification.

3.1. Advantages

- Less prone to overfitting and works on Bootstrapped sampling and works better for regression analysis.

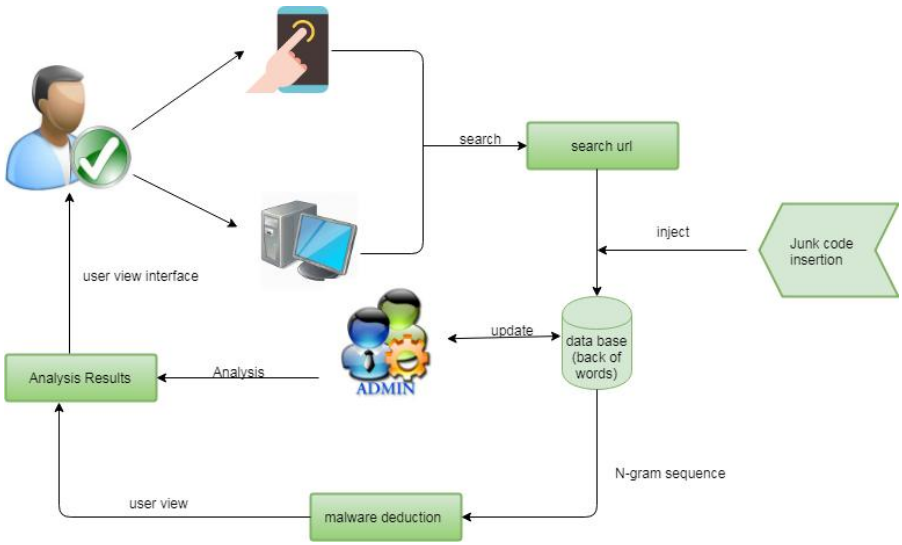

Figure 1. Proposed malware detection architecture.

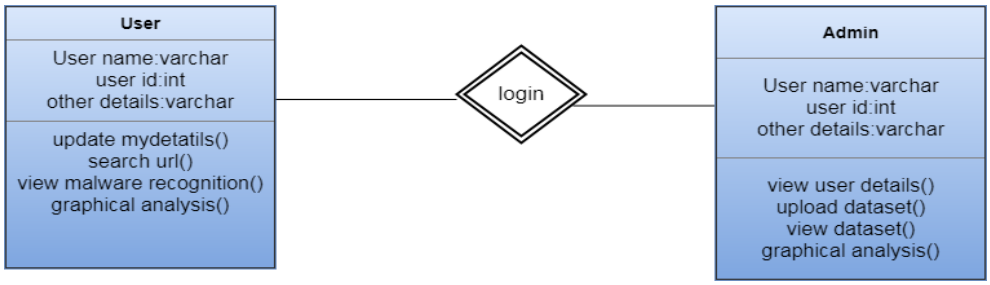

Figure 2. Class diagram of proposed malware detection.

3.2. MODULES

There are three modules can be divided here for this project they are listed as below**:**

- User Activity.
- Malware Deduction**.**
- Junk Code Insertion Attacks**.**

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.

Malware Deduction: Users search the any link notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malwares 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).

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.

4. CONCLUSIONS

IoT, particularly IoBT, will be increasingly important in the foreseeable future. No malware detection solution will be foolproof, but we can be certain of the constant race between cyber attackers and cyber defenders. Thus, it is important that we maintain persistent pressure on threat actors. In this paper, we presented an IoT and IoBT malware detection approach based on class-wise selection of Op-Codes sequence as a feature for classification task. A graph of selected features was created for each sample and a KNN and random forest learning approach was used for malware classification. Our evaluations demonstrated the robustness of our approach in malware detection with comparison to the existing KNN classifier.

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] F. Leu, C. Ko, I. You, K.-K. R. Choo, and C.-L. Ho, "A smart phone-based wearable sensors for monitoring real-time physiological data," Computers & Electrical Engineering, 2017.

[3] 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.

[4] 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.

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

[6] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7, pp. 1497–1516, 2012.

[7] A. Kott, A. Swami, and B. J. West, "The internet of battle things," Computer, vol. 49, no. 12, pp. 70–75, 2016.

[8] M. J. Farooq and Q. Zhu, "Secure and reconfigurable network design for critical information dissemination in the internet of battlefield things (iobt)," arXiv preprint arXiv:1703.01224, 2017.

[9] C. Tankard, "The security issues of the internet of things," Computer Fraud & Security, vol. 2015, no. 9, pp. 11 – 14, 2015.

[10] E. Bertino and N. Islam, "Botnets and internet of things security," Computer, vol. 50, no. 2, pp. 76–79, Feb 2017.
[11] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," ACM Computing Surveys, vol. 49, no. 3, p. Article No. 59, 2016.
[12] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," Journal of Network and Computer Applications, vol. 72, pp. 14–27, 2016.
[13] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "Iot security: Ongoing challenges and research opportunities," in 2014 IEEE 7th International Conference on Service Oriented Computing and Applications, Nov 2014, pp. 230–234.
[14] Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrows intelligent network traffic control systems," IEEE Communications Surveys & Tutorials, 2017.
[15] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided android malware classification," Computers & Electrical Engineering, 2017.
[16] K. Shaerpour, A. Dehghantanha, and R. Mahmod, "Trends in android malware detection," The Journal of Digital Forensics, Security and Law: JDFSL, vol. 8, no. 3, p. 21, 2013.
[17] Z. Bazrafshan, H. Hashemi, S. M. H. Fard, and A. Hamzeh, "A survey on heuristic malware detection techniques," in Information and Knowledge Technology (IKT), 2013 5th Conference on. IEEE, 2013, pp. 113–120.
[18] O. E. David and N. S. Netanyahu, "Deepsign: Deep learning for automatic malware signature generation and classification," in Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, 2015, pp. 1–8.
[19] R. Pascanu, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, "Malware classification with recurrent networks," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 1916–1920.

