International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *16 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

Exploiting And Estimating Malware Using Feature Impact Derived From API Call Sequence Learning

M.Sunitha Reddy[1], Rupa Devi T[2], Sirisha K L S[3], Soma Sekhar G[4], VDS Krishna[5], Pradeep Kumar V[6]
[1]Department of CSE
Vasavi College of Engineering
Hyderabad, India
m.sunithareddy@staff.vce.ac.in
[2]Department of CSE
Keshav Memorial Institute of Technology
Hyderabad, India
rupa.devi179@gmail.com
[3]Department of CSE
Keshav Memorial Institute of Technology,
Hyderabad, India
klssirisha@kmit.in
[4]Department of CSE
Vardhamaan College of Engineering
Hyderabad, India
somasekharonline@yahoo.co.in
[5]Department of CSE
CVR College of Engineering
Hyderabad, India
vds.krishna@cvr.ac.in
[6] Department of CSE
B V Raju Institite of Technology
Medak, India
pradeepkumar.v@bvrit.ac.in

Abstract—Malware is a serious threat being posed and it has been a continuous process of protecting the systems from existing and new malware variants by defining new approaches for malware detection .In this process malware samples are first analyzed to understand the behavior of the vulnerable samples and accordingly statistical methods are defined for malware detection. Many approaches are defined for understanding the behavior of malware executables which are broadly classified in to static and dynamic assessments. The static analysis can only be used for identifying the existing types of malware but code obfuscation has made it complex to identify the variants of existing malware. To counter the code obfuscation the dynamic analysis of malware is prioritized over static analysis where the malwares are analyzed by running them in an emulated environment to understand the intent of the samples. As there is an acute need of developing a more precise and accurate approach for malware detection, this paper contributes in the above said direction where we proposed a novel measure to estimate malware by exploiting the malicious intent of executables. It is a machine learning approach where the knowledge is acquired from the existing malicious executable and the same knowledge is used to estimate the new variants of the existing malware. The proposed statistical approach can be used to improve the scalability, accuracy and robustness. It also defends against zero day exploits.

Keywords- Binary Executables , Benign, Malign, Obfuscation, Zero day threats, Exploratory Scale.

I. INTRODUCTION

All manuscripts must be in English. These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts. Please follow them and if you have any questions, direct them to the production editor in charge of your proceedings at Conference Publishing Services (CPS): Phone +1 (714) 821-8380 or Fax +1 (714) 761-1784.

This template provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *16 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

throughout this document and are identified in italic type, within parentheses, following the example. PLEASE DO NOT RE-ADJUST THESE MARGINS. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. RELATED WORK

Many methods were proposed for malware detection and the majority of them used features extracted from API. The method proposed by Mamoun Alazab et al. [7] is also based on features extracted from the APIs which performs analysis of the features automatically for identifying the hidden malicious actions. Bot traffic that can be used to find command and control connections is being proposed in [8]. Machine learning methods for identifying and separating the benign and malicious samples was proposed in [9]. An in-depth look at semantic features like the programme instruction matrix and address references was described in [10].

Detection of malware using graphical representation of instruction traces is proposed in [11], [12] where in the graph each instruction is shown as a vertex and multiple graphs are combined for developing the best similarity matrix among the graphs which is in turn used for classification. In [13], n-grams from features and API are derived. SVM was used to model API features, and NN was used to model n-grams. Later, D-S theory is used to combine the results of classifiers. There was an accuracy rate of 98.73 percent based on this. malevolent programmers have been around for a while, have been making malware that automatically makes more copies of itself. The properties of a few malware variants from the same family frequently overlap. Nevertheless, in order to identify and categorize malware from various families, [14] produced a picture out of the intrusion code. After that, the image was divided into graph entropy. To compare the similarity of two graphs, it used the similarity of updated histograms [15]. The models' accuracy was very high, however they were not very efficient.

API call sequences are used in the proposed method where features are n gram call sequences. Since there is some noise in the call sequence, the call sequence similarity comparison is only roughly accurate and frequently fails. Here we prefer 2 gram sequences as features In order to reduce the impact of call sequence noise, our proposed scale chooses 2 gram sequences as features as opposed to the call sequence similarity comparison, which favors n-gram sequences. As a result, our suggested approach can detect malicious sequences when there is noise influence.

III. MEASURING THE FEATURE IMPACT

In the process of using API call sequencing for malware detection, we extracted 2 gram sequences as features and then the best of the features are identified using sequential floating search method[17][18]. Afterward, the effects of each feature and call sequence of the specified malicious and benign samples are measured using bipartite graphs [14][16]. The identified impact values can be used to judge whether a sample is malicious or benign. The given data set is partitioned in to two sets based on their Boolean values where one set contains the call sequence records which are benign and other set contains those records which are malicious. Then this data set is used for measuring the impact of the features for judging whether a new sample is malicious or benign.

A. Using API Call sequences

In order to measure the Feature impact the training data set containing the call sequences is partitioned into two sets MC=mc1,mc2,mc3. . . . . mcn and
BC=bc1,bc2,bc3. . . . . bcn
Now each two distinct consecutive calls are treated as features and they are represented as MF and BF
MF= mf1,mf2,mf3,mf4. . . . . mfn are the features found in MC where we treated two unique calls in sequence as malicious features BF= bf1,bf2,bf3,bf4. . . . . bfn are the features found in BC where we treated two unique calls in sequence as benign features.

B. Optimal Feature Selection

Sequential floating search method is used for selecting the optimal features where the size of best feature set increases as new features are added and it decreases when worst of it are removed. A feature is considered as best feature when it has majority appearances in MC and minimum appearances in BC. Similarly, a feature is considered as worst feature if it has majority appearances in BC and minimum appearances in MC. The Sequential floating search method removes the worst features.

C. Measuring the Feature confidence and Call sequence confidence

In order to measure the feature impact first we need to measure the feature pair correlation which helps in identifying the confidence of a feature towards a call sequence and then this confidence value is used to measure the confidence of every feature characteristic with respect to a call sequence of MC and BC. It also helps in measuring the confidence of each call sequence. Finally, the effectiveness of each feature in malicious and benign sets is evaluated using the feature confidence and call sequence confidence.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *16 August 2023* ***Revised:*** *12 October 2023* ***Accepted:*** *20 October 2023*

1) Measuring Feature pair correlation :

By dividing the number of harmful call sequences that contain both features by the overall number of malicious call sequences, one may determine the correlation between the feature pairs. Similar to this, the feature pair correlation for each pair of two calls in a sequence will be calculated as the proportion of benevolent call sequences that contain both characteristics to all benevolent call sequences.

2) Measuring feature to call sequence confidence :

The sum of the correlation between each pair of features that contain a feature divided by the total number of features is the feature confidence of each call sequence of MC and BC.

3) Measuring call sequence confidence:

The call sequence confidence of MC and BC is the sum of the confidence of all the features found in MC divided by the total number of features found in MC. This is how it is measured.

In a similar vein, the confidence for every call sequence in BC is calculated by adding the confidence of each feature toward each individual call sequence and dividing the result by the overall number of features identified in BC.

4) Measuring feature impact:

The difference between the aggregated confidence of call sequences that contain a given feature and the aggregated confidence of all call sequences discovered in MC is now used to calculate the feature effect of MC and BC.

The aggregate confidence of call sequences that include each individual feature is divided by the aggregate confidence across all call sequences discovered in BC to get the feature impacts of BC.

5) Measuring call sequence impact:

The impact of call sequence of every call sequence in MC is calculated as the sum of all the features that are present in the particular call sequence divided by the sum of all the features present in MF.

Similar to this, the aggregate impact of all features found in each call sequence of BC can be calculated by dividing the aggregate impact of all features present in every call sequence by the aggregate impact of all features present in BF.

IV. EXPERIMENTAL STUDY

The experimental research was conducted on CSDMC2010 API [15] which contained 388 call sequences which are mentioned as 1 (malign call sequence) and 0(benign call sequence). As to measure the feature impact, 70% of malign and benign call sequences of the total dataset is used. The rest 30% call sequences are not labeled and used to measure the accuracy of detecting the malware. Feature impact and call sequence impact can be used to detect malware, and the results are shown in Table 1.

Malicious Call Sequence	90
Benign Call Sequence	27
True Positives	89
True Negatives	24
False Positives	3
False Negative	1
Accuracy	0.98988
Sensitivity	0.9888
Specificity	0.8888

Table 1
PREDICTION STATISTICS

Based on all 117 call sequences, the projected ESMP's importance was evaluated. There are 27 benign call sequences and 90 detrimental call sequences among them. 89 call sequences from the input call sequences evaluated by the suggested approach are true positives. 1 call sequence was false positive, 24 call sequences were true negatives, and 3 call sequences were false negatives. Hence the sensitivity obtained is 0.98 and specificity obtained is 0.88 and hence the accuracy is 0.98. These statistics show that the proposed method could be used to get a 98 percent rate of success with it.

V. CONCLUSION

What the model does is as follows: The training set of harmful and benign call sequences is examined using the two-gram call sequences as features. Then, a sequential floating forward search is used to identify the best characteristics. The influence of these top attributes and associated call sequences are then utilized to search for malware.The experimental study shown the accuracy as 97%. This work can be extended to define the exploratory scale for malware detection.

REFERENCES

[1] E. K. M. Egele, T. Scholte and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," ACM Computing Surveys (CSUR), vol. 44, 2012.

[2] L. J. Y. Qiao, Y. Yang and J. He, "Analyzing malware by abstracting the frequent itemsets in api call sequences," In Proc. of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'13), Melbourne, pp. 265–270, 2013.

[3] H. K. Y. C. B. Kang, T. Kim and E. G. Im, "Malware classification method via binary content comparison," In Proc. of the 1st 2012 ACM Research in Applied Computation Symposium (RACS'12), New York, USA, pp. 316–321, 2012.

[4] P. C. A. H. Sung, J. Xu and S. Mukkamala, "Static analyzer of vicious executables (save)," In Proc. of the 20th Computer Security Applications Conference (ACSAC'04), Tucson, AZ, USA, pp. 326–334, 2004.

[5] T. H. C. Willems and F. Freiling, "Toward automated dynamic malware analysis using cwsandbox," ACM Computing Surveys (CSUR), pp. 32–39, 2007.

[6] G. Hunt and D. Brubacher, "Detours: Binary interception of win32 functions," In Proc. of 3rd USENIX Windows NT Symposium (WINSYM'99), pp. 135–144, 1999.

[7] S. V. M. Alazab and P. Watters, "Towards understanding malware behaviour by the extraction of api calls," In Proc. of the 2nd Cybercrime and Trustworthy Computing Workshop (CTC'10), pp. 52–59, 2010.

[8] Jacob and Gregoire, "Jackstraws: Picking command and control connections from bot traffic," USENIX Security Symposium, 2011.

[9] Demme and John, "On the feasibility of online malware detection with performance counters," ACM SIGARCH Computer Architecture News, 2013.

[10] Ozsoy and Meltem, "Malware-aware processors: A framework for efficient online malware detection," High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on.IEEE, 2015.

[11] P. A. M. Al-Bakri and H. L, "Static analysis based behavioral api for malware detection using markov chain," The International Institute for Science, Technology and Education (IISTE), 2014.

[12] Anderson and Blake, "Graph-based malware detection using dynamic analysis," Journal in Computer Virology, pp. 247–258, 2011.

[13] G. B. Krishna, V. Radha, and K. V. Rao, "Esmp: Exploratory scale for malware perception through api call sequence learning," Journal of Theoretical and Applied Information Technology, vol. 95, pp. 2297–2305, 2017.

[14] G. B. Krishna, V. Radha, and V. Rao, "Elc-ppw: Ensemble learning and classification (lc) by positional patterns weights (ppw) of api calls as dynamic n-grams for malware perception," International Journal of Simulation: Systems, Science and Technology, vol. 18, 2018.

[15] V. R. G Bala Krishna, Vedala Radha, "Evolutionary binary classification using cuckoo search for malware perception in api call sequences," IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, 2017

[16] Rani, D. S. ., Krishna, G. B. ., Radha, M. ., K L S, S. ., Kumar, V. P. ., & Anupama, M. . (2023). GBJOF: Gradient Boosting Integrated with Jaya Algorithm to Optimize the Features in Malware Analysis . International Journal on Recent and Innovation Trends in Computing and Communication,11(8s)

[17] Gn, Balaji & Suryanarayana, S & Veeramani, C.. (2018). Invariant Hand Gesture Recognition System. International Journal of Engineering and Technology(UAE). 7. 299-301. 10.14419/ijet.v7i4.6.20717.

[18] K., L. P. ., Suryanarayana, G. ., Swapna, N. ., Bhaskar, T. ., & Kiran, A. . (2023). Optimizing K-Means Clustering using the Artificial Firefly Algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(9s)