International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Predicting the Dynamic Behaviour of Malware using RNN

Anuradha Sengupta, S. Sivasankari, V. Joseph Raymond

Abstract: ***Malware analysis can be classified as static and dynamic analysis. Static analysis involves the inspection of the malicious code by observing the features such as file signatures, strings etc. The code obfuscation techniques such as string encryption, class encryption etc can be easily applied on static code analysis. Dynamic or behavioural data is more difficult to obfuscate as the malicious payload may have already been executed before it is detected. In this paper, the dataset is obtained from repositories such as VirusShare and is run in Cuckoo Sandbox with the help of the agent.py. The dynamic features are extracted from the generated Cuckoo logs in the html and JSON format and it has to be determined whether it is malicious or not using recurrent neural networks. Recurrent Neural Networks are capable of predicting whether an executable is malicious and have the ability to capture time-series data.***

Keywords: Behavioural Data, Cuckoo Sandbox, Recurrent Neural Networks, Zero-day Malware.

I. INTRODUCTION

The number of malwares being discovered everyday is increasing and an automated detection system is required to detect these malwares. VirusTotal is an API which can be used to detect whether a file is malicious or not based on file signatures, number of anti-virus engines detecting the file as malicious etc. The zero-day malware cannot be detected this way if it does not share any code with any malware previously detected. Dynamic or behavioural analysis is defined as the process of executing the malware sample in a virtual environment and analysing and recording the behaviour of the malware. The malware while executing may drop files and leave other footprints while executing, therefore allowing the malware analysts to understand the malicious payloads in a better manner. The execution of the malware may take a long time. The anti-virus engines do not often use the dynamic data due to the amount of time needed to execute the malware and observe its features and behaviour. To avoid waiting, the live features and activities of the malware is monitored. The monitoring and detection systems observe any deviations from the normal behaviour of a malware from a baseline. In this paper, the datasets are downloaded from the dataset repositories such as VirusShare, Virus Total etc and is run in Cuckoo Sandbox.

Revised Manuscript Received on February 05, 2020.

* Correspondence Author

Anuradha Sengupta*, Department of Information Technology, SRM Institute of Science and Technology, Chennai, India. Email: anuradhasengupta.as@gmail.com

S. Sivasankari, Department of Information Technology, SRM Institute of Science and Technology, Chennai, India Email: sivasan2@srmist.edu.in

V. Joseph Raymond, Department of Information Technology, SRM Institute of Science and Technology, Chennai, India. Email: josephrv@srmist.edu.in

The dynamic features are extracted from the cuckoo reports and logs and has to be given as an input to the RNN and then it is predicted whether it is malicious or not.

II. REVIEW OF LITERATURE

There are many research works which have been carried out in the field of static and dynamic detection of malware. Each of these research works provide an insight into the numerous tools, techniques, frameworks etc.

Smita Ranveer et al. [1] had done a comparative analysis of all the malware detection systems and the features extracted by each of these systems.

S. L. S. Darshan et al. [2] have run the various malicious payloads in Cuckoo sandbox and have extracted the system calls from the analysis reports.

Chaudhari et al. [3] have studied about the extraction of features from the malware file. The authors have used tools such as VirusTotal, IDA Pro etc.

Si .N et al. [4] has proposed a classification algorithm based on static features called MCSC and identifies malware families by CNN.

Burnap P et al. [5] proposes a method for the automatic detection of malicious and benign malware executables sing machine activity features such as CPU, RAM etc.

Catak et al. [6] have analysed and run various malicious payloads in a sandboxed environment and have used the results to classify the various categories of malware such as virus, ransomware etc.

Zhou H. [7] has proposed a framework for using both static and dynamic features for determining whether a PE file is malicious or not.

R.Vinayakumar et al. [8] have designed a framework called ScaleMalNet. The authors have used static, dynamic and image-processing techniques for detecting and classifying zero-day malware.

Ijaz et al. [9] discusses about static and dynamic analysis methods of malware using various algorithms and also the limitations of dynamic analysis due to a controlled network environment.

S. Mohammed A. F et al. [10] discusses about the increasing complexity of malware and behaviour of the execution of a PE file. The authors also discuss about n-gram technique.

III. EXISTING SYSTEM

The static malware analysis deals with the analysis of the code of the malicious file and checking the file signatures, strings, opcodes etc. The static analysis can be done using tools such as PEiD, Dependency Walker, PEView, Virus Total etc.

The static analysis cannot be done effectively if the code is obfuscated and the malware developer has obfuscated the strings, API calls, methods etc. Therefore, the dynamic analysis needs to be done to track the behaviour of the malware during its runtime execution.

IV. PROPOSED SYSTEM

In the proposed system as shown in Fig. 1, the malware executables are downloaded from data repositories such as VirusShare and the malware executables are extracted. The VirusTotal API is then used to determine the properties of the executable such as the file signatures, the number of anti-virus engines which have declared it as a malware etc. The malware executables are the run in Cuckoo Sandbox and the dynamic features are extracted. The dynamic features then have to be given as an input to the Recurrent Neural Networks. The RNN will help in predicting whether executable is malicious or not.

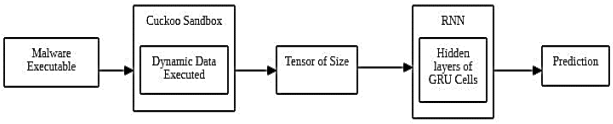

Fig. 1. System Architecture of Proposed System

V. EXPERIMENTAL RESULTS

A. Downloading of dataset and uploading in VirusTotal

The datasets are downloaded from data repositories such as VirusShare and the malware executable is uploaded in VirusTotal API as shown in Fig .2 to check the number of anti-virus engines detecting as malicious, the file signatures, the strings etc.

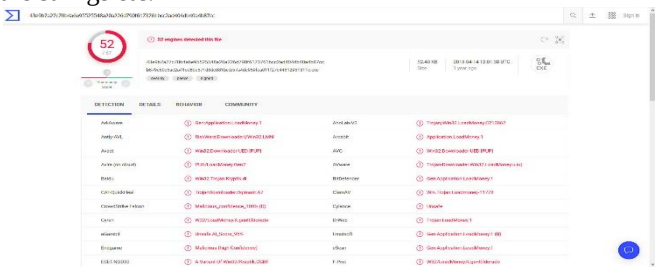

Fig. 2. Uploading of the malware executable in VirusTotal

B. Cuckoo Sandbox Architecture

The Cuckoo sandbox is a malware analysis tool and can be used to detect an analyse any suspicious file. The tool can analyse files of different types such as executables, URLs etc. It can also trace API calls, capture network traffic in the PCAP format, perform memory analysis using Volatility etc. The Cuckoo sandbox architecture consists of the Cuckoo host, the virtual network and the analysis virtual machines. After the analysis is done the reports are given back to the Cuckoo host as shown in Fig. 3.

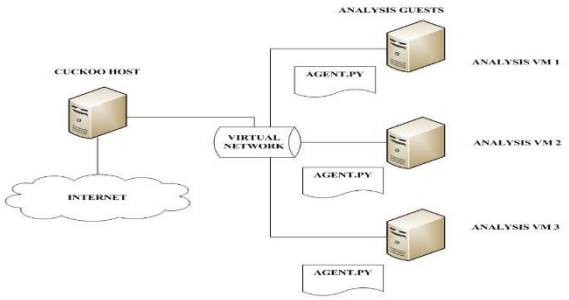

Fig. 3. Cuckoo Sandbox Architecture

C. Executing Malware Executable in Cuckoo Sandbox

The cuckoo.py program is executed and the server is run. The malware is submitted to the cuckoo web API and then the analysis is started as shown in Fig. 4. The virtual machine is restarted as shown in Fig. 5 and the malware executable is run in the clean virtual environment with the help of the agent.py as shown in Fig. 6. The reports and results are reported back to the Cuckoo host upon execution of the malware in html and JSON format as shown in Fig. 7 and Fig. 8 respectively.

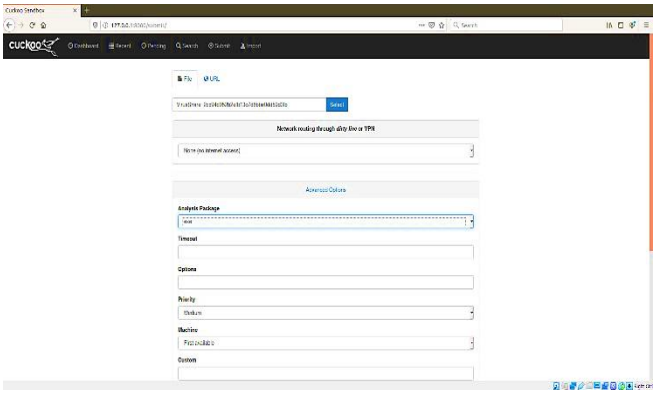

Fig. 4. Submitting of the malware executable in Cuckoo Web

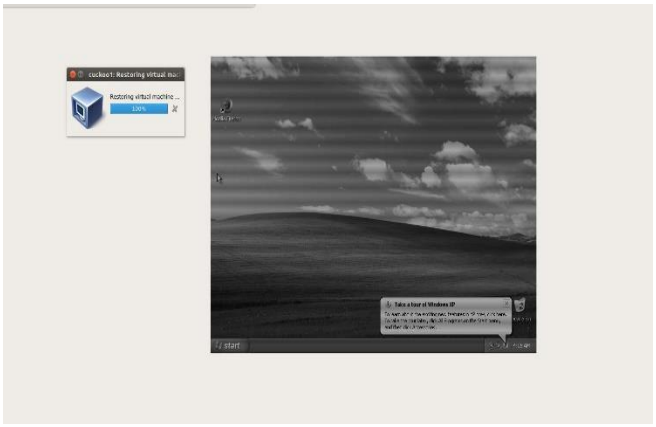

Fig. 5. Restoring of the Virtual Machine after executable is submitted

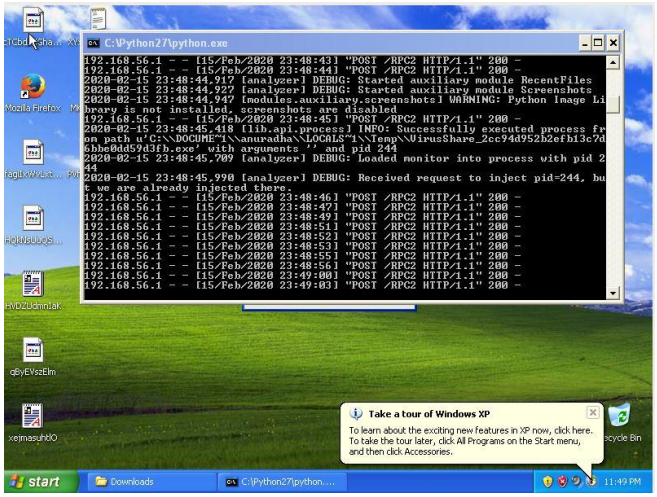

Fig. 6. Malware executable executing in Cuckoo Sandbox with the help of agent.py

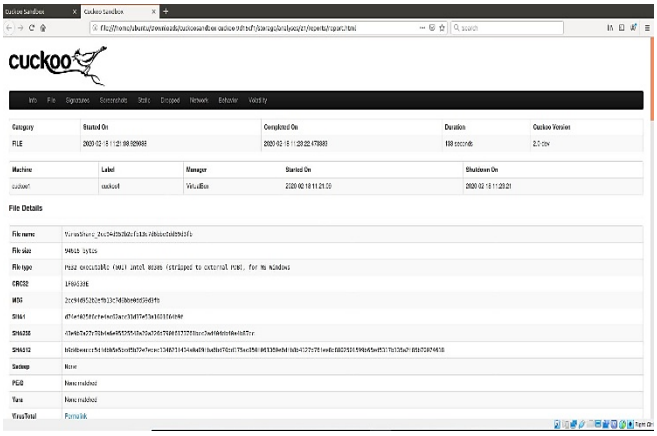

Fig. 7. Cuckoo reports in html format

Fig. 8. Cuckoo reports in JSON format

D. Extraction of Dynamic Features from Cuckoo reports and logs

The Cuckoo reports and logs in the html and JSON format can be used for extracting the dynamic features as shown in Fig. 7 and Fig. 8 respectively. The dynamic features can be the duration, the size of the malicious executable, the dropped files, the files read and opened, the registry keys added etc. These dynamic features have to be given as an input to the Recurrent Neural Networks (RNN). The RNN will help in predicting whether the executable is malicious or not.

VI. CONCLUSION AND FUTURE WORKS

In this paper, the malicious executables have been downloaded from data repositories such as VirusShare and is run in Cuckoo Sandbox with the help of agent.py. The dynamic features are extracted from the Cuckoo reports and logs in the html and JSON format. In the future work the dynamic features have to be given as an input to the RNN and the executable has to predicted whether it is malicious or not. Similarly, the other file formats such as .pdf, .doc etc can be executed in Cuckoo sandbox for prediction of any malicious activities.

REFERENCES

1. Smita Ranveer and Swapnaja Hiray.Article: Comparative Analysis of Feature Extraction Methods of Malware Detection. International Journal of Computer Applications 120(5):1-7, June 2015.
2. S.L.S Darshan, M.A.A Kumara and C.D. Jaidhar, “Windows malware detection based on cuckoo sandbox generated report using machine learning algorithm,” 2016 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee,2016, pp.534-539.
3. Chaudhari, Hemant & Guru, Shri & Singhji, Gobind & Mahindrakar, Manisha Mahindrakar & Guru, Gobind & Singhji, (2017). Noble Feature Extraction of Malware from Contents of File. 10.13140/RG.2.2.1794445.21601.
4. S. Ni, Q. Qian and R. Zhang, “Malware Identification using visualization images and deep learning,” Comput. Secur. , vol. 77, pp.871-885, Aug. 2018.
5. Burnap P, French R. Turner F, Jones K. Malware classification using self organising feature maps and machine activity data. ComputSecur 2018; 73:399-410.
6. Catak, Ferhat Ozgur &Yazi, Ahmet. (2019), A Benchmark API Call Dataset for Windows PE malware Classification.
7. Zhou H. (2019) Malware Detection with Neural Network using Combined Features. In: Yun X. et al. (eds) Cyber Security. CNCERT 2018. Communications in Computer and Information Science, vol 970.Springer, Singapore.
8. R. Vinayakumar & Alazab, Mamoun & Kp, Soman & Poornachandran, Prabaharan & Venkatraman, Sitalakshmi. (2019). Robust Intelligent Malware Detection Using Deep Learning. IEEE Access. PP.1-1.10.1109/ACCESS.2019.2906934.
9. Ijaz, Muhammad & Durad, Hanif &Ismail, Maliha. (2019). Static and Dynamic Malware Analysis Using Machine Learning.687-691.10.1109/IBCAST.2019.8667136.
10. S. Mohammed A.F., M.F. Marhusin and R. Silaiman, “Instrumenting API Hooking for a Realtime Dynamic Analysis,” 2019 International Conference on Cybersecurity (ICoCSec), Negeri Sembilan, Malaysia, 2019, pp. 49-52.

AUTHORS PROFILE

Anuradha Sengupta is a postgraduate student currently studying Masters in Information Security and Cyber Forensics from SRM Institute of Science and Technology, Chennai, India.

S. Sivasankari is an Assistant Professor at SRM Institute of Science and Technology, Chennai, India.

V. Joseph Raymond is an Assistant Professor at SRM Institute of Science and Technology, Chennai, India.