Malware Analysis using GPGPU

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,*
and Jane Doe, *Life Fellow, IEEE*

Abstract—The abstract goes here.

Index Terms—Malware Detection, GPGPU, Parallel Naive Bayes, Machine Learning, Computer Security, CUDA.

1 Introduction

Malware refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victims data, applications, or operating system or of otherwise annoying or disrupting the victim [1]. There has been a significant development in the field of malware creation in terms of writing highly complex and undetectable malwares and based on the complexity of malwares they can be categorized as first generation and second-generation malwares. In first generation, structure if malware does not change, while in the second generation, structure changes to generate new variant, keeping the action same [2]. On the basis of how variances are created in malware, second generation malware are further classified into Encrypted, Oligomorphic, Polymorphic and Metamorphic Malware [3]. According to 2017 Threat analysis report they has now more than 780 million malware samples in their database and the new malwares increased by 10% in third quarter to 57.6 million. In addition to it 60% increase in new mobile malware is also observed in the third quarter of 2017 due to large increase in Android screen locking ransomware [4]. The Symantec 2017 Internet security Threat report indicates that there were 357 million new malware variants, 3.6 thousand new mobile malware variants were detected and every two minutes an IoT device is being attacked [5]. This increase in malware threat can be corelated to the increasing use of worldwide web due to the exponentially increasing mobile and IoT devices resulting increased attack surface. The malware attack/threat are not only limited to individual boundaries, but they are highly skilled state funded hackers writing customized malicious payloads to disrupt political, industrial working and military espionage [5][6]. The most high-profile, subversive incident of the year was a series of intrusions against the Democratic Party, which occurred in the run-up to the 2016 US presidential election [5].

It is indubitable fact that various traditional (signature based) approaches are ineffective to combat the dynamic and complex behavior of second generation viruses. If adequate advancement in anti-malware techniques are not achieved, consequences at this scale (more than 56.6 million new viruses are reported in one quarter year [4] in 2016) at which new malwares are being developed can create fatal affects and the results will we more severe then past as due to more reliance on digital world. The second-generation malwares contain very complex structure and advanced obfuscation techniques to make the detection process harder and counter the treat. Recently in 2107 the WannaCry ransomware drew the attention of researchers from all over world causing a lot of trouble. Another major threat I CCleaner was disarmed before it would have

- *M. Shell is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: see http://www.michaelshell.org/contact.html*
- *J. Doe and J. Doe are with Anonymous University.*

caused any widespread harm, despite an estimated 2.27 million infections. Therefore, there is a need that researchers and anti-malware developers should work side by side to counter the threat/attack from the new malwares. The most widely used malware detection engines are based on signature-based detection, malware normalization, heuristic based detection, machine learning, etc. [3].

In recent years, various machine learning techniques has been proposed by authors [8][9][10][11], which can enhance the capabilities of traditional malware detection engines(signature based detection) but, with the use of a complex machine learning based anti-malware engine the detection time increases. This can result in inefficient resource utilization and less throughput hampering the performance of whole system. Hence, in this paper we discuss an approach to detect malwares with high throughput and accuracy. We present a static malware analysis technique using GPGPU for faster detection of malwares based on the approach proposed by authors in [10]. We were able to achieve a maximum speedup of 120 over the actual implementation [10] and achieving the same accuracy mentioned by authors. The remaining work is organized as follows. Section 2 discuss about the related work, in section 3 we discuss the dataset description, data preprocessing and feature selection. Section 4 contains the brief description of the Nave Byes and detection technique. Section 5 describes proposed approach for parallel implementation of Nave Bayes using CUDA architecture. Finally in section 6 contains conclusion and future scope of work.

2 Related Work

yet to write

3 Data Preprocessing and Feature Selection

3.1 Dataset

For this experiment we downloaded 11355 malware samples from malacia-project and collected 2967 benign programs (also verified from virustotal .com) from different systems. In the collected dataset it was found that majority of malwares are below 500 KB hence for our study we focused on malware and benign files below 500 KB. After applying the size limit of 500 KB on samples we are left with 2363 benign samples and 11305 malware samples for our work.

3.2 Preprocessing

The malware and benign samples were processed using objdump utility to get opcodes. A unique opcode list was prepared from the processed samples which was then used to make the feature matrix for each malware or benign sample. For each sample the feature matrix consists of the frequency of a particular opcode in a malware and benign sample. The data was normalized by dividing each malware and benign opcode frequency by corresponding maximum opcode frequency.

3.3 Feature Selection

To find distinctive features in a group we first divide the normalized frequencies of malware and benign per group by the total number of malwares and benign in that group and then perform column wise sum for that group where each column denotes an opcode. By performing the above procedure, we get two different vectors corresponding to malware and benign. Now we find absolute difference between the malware and benign vector where each column entry corresponds to an opcode. The absolute difference is then sorted in descending order. Finally, top K features (opcodes) are chosen based on the top K absolute difference. The whole process is done to find those opcodes which are able to separate malware and benign clearly in group. This process is then repeated for each group. It was observed that some of the groups were not having sufficient malware or benign samples for training and testing so we neglected them from our course of study (group 5,8,61,65,97 contain less than 6 samples in either malware or benign classes while group 98 and 100 has 0 malware samples). However, with more malware and benign samples in hand it they can be included. The preprocessed data is the divided into testing and

training sets. We used 67% data for training and reaming 33% data for testing.

4 Naive Bayes Classifier

5 Proposed approach

6 Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgments

The authors would like to thank...

References

[1] P. Mell, K. Kent, J. Nusbaum, Guide to malware incident prevention and handling, US Department of Commerce, Technology Administration, National Institute of Standards and Technology, (2005).

[2] Govindaraju, Exhaustive statistical analysis for detection of metamorphic malware, Masters thesis, San Jose State University (2010).

[3] A. Sharma, S. K. Sahay, Evolution and detection of polymorphic and metamorphic malware: A survey, International Journal of Computer Applications., vol. 90, no. 2, (2014), pp 7-11

[4] Niamh Minihane, Francisca Moreno, Eric Peterson, Raj Samani, Craig Schmugar, Dan Sommer , Bing Sun, McAfee Labs Threat Report, December 2017.

[5] Kavitha Chandrasekar Gillian Cleary Orla Cox Hon Lau Benjamin Nahorney Brigid O Gorman Dick OBrien Scott Wallace Paul Wood Candid Wueest, Internet security threat report 2017, Symantec Corporation, USA, (2017).

[6] R. Stone, A call to cyber arms, Science., vol. 339, no. 6123, (2013), pp. 10261027.

[7] Robert M. Lee, Michael J. Assante, Tim Conway, Analysis of the Cyber Attack on the Ukrainian Power Grid, E-ISAC group SANS, March 2016.

[8] Allix, K., Bissyande, T.F., Jerome, Q., Klein, J., Le Traon, Y., et al.: Large-scale machine learning-based malware detection: confronting the 10-fold cross validation scheme with reality. In: Proceedings of the 4th ACM conference on Data and application security and privacy, ACM (2014) 163166.

[9]] Canto, J., Dacier, M., Kirda, E., Leita, C.: Large scale malware collection: lessons learned. In: IEEE SRDS Workshop on Sharing Field Data and Experiment Measurements on Resilience of Distributed Computing Systems. (2008)

[10]] A. Sharma, S. K. Sanjay, Improving the detection accuracy of unknown malware by partitioning the executables in groups, Proceedings of the 9th international confrence on advanced computing and comunication technologies, Nagpur, India, (2015).

[11] Daniele Uccia, Leonardo Aniellob, Roberto Baldonia, Survey on the Usage of Machine Learning Techniques for Malware Analysis, arXiv:1710.08189v2 [cs.CR], Feb 2018.