Malware Detection Using Integrated Static and Dynamic Analysis.

FAYAZ KAMALI, UNIVERSITY OF BRADFORD.

An alarming trend in malware attack's are that they are equipped with advance sophisticated techniques to breakout from security countermeasures. Detecting a malware is challenging but combining different detection models together increases the chances by exploring malicious activity from different viewpoints, this paper proposes a collaborative static and dynamic analysis for spotting malicious behaviors of files. Malware detector is the essential tool in protecting against malware's. The results of such detector are determined by the methodologies it uses. therefore, it is essential to investigate malware detection techniques and identify their strengths and boundaries.

1. INTRODUCTION

Internet is becoming the essential part of our daily life; the drastic increase of evolving malware is threatening for today's digital world and to the society itself. Recent Malware's are becoming more sophisticated and evasive therefore it's very hard to detect changing nature of malwares, According to F-secure only 15-20 malware blocked per 10000 users (Vasilescu et al. 2014) and according to AV-Test over 390,000 malwares are created everyday (Shijo and Salim 2015) , Detection and prevention of malware is a global challenge; statistics indicates that the impact of malware is getting worse and eviler, it is essential to develop new approaches to filter and detect malicious content and discover new tactics that combines different malware analysis techniques to get better results. Malwares are not only limited to steal private information but also they could infect the host to form botnet which then used to take part in Distributed Denial of Services (DDoS) attacks (Shijo and Salim 2015), Commonly used technique by current antiviruses to detect malware is signature based detection, which look's for known malicious behaviors within suspicious files, Signature based detection match's malicious files against already defined signature pattern's, this requires regular updates of the signature database which is one of the disadvantage, while Dynamic analysis checks the malicious actions in executions to discover whether the executable is malicious or not , Both static and dynamic analysis have its own advantages and disadvantages. This paper proposes an integrated method that utilizes both features to analyses unknown malicious files.

2. BACKGROUND AND RELATED WORK.

In this segment, various different types of malware detection and their shortcomings are questioned when used as a single analysis for malware detection, In order to get better results only using one method is not sufficient to detected evasive malwares.

2.1 Malware Analysis

By exploring a malware, one can figure out vital useful details such as, IPs of originating servers, illegitimate file access, whether packer's software's used or not, if it has been obfuscated, whether it's distributed on multiple platforms or not. All this information can help investigator to find out the impact of the attack, whether the hacker is an individual or an organized cyber-crime group. To investigate malwares number of methods are available to perform malware analysis, for example static analysis, dynamic analysis, memory analysis, honeypots and automatic analysis.

2.2 Static Analysis

As the name indicates, static analysis collects information about the malware's without executing it. Using dissembled version static analysis fetch public string information and examine packers, However, it fails at sophisticated code obfuscation used by attacker and also at polymorphic and metamorphic malwares (Saxena et al. 2015). One of the major problem in examining malwares are the freely available tools that alter an executable file's to obfuscate its content and hide the actual program logic,

Programs that alter other program files to disguise their contents are called packers therefore once the program is been packed the original logic of the program is hard to recover through static analysis techniques . When performing static analysis you are actually conducting an autopsy of the code at rest state which fails at sophisticated obfuscation , polymorphic and metamorphic malwares (Moser et al. 2007).

2.3 Dynamic Analysis

Dynamic analysis examines the malware's communication with host at runtime, it includes probing malware interaction with file system, network processes, etc. it requires isolated environment where the activities of malwares are monitored. Dynamic analysis detects code obfuscations and polymorphic malwares because it investigates the program while in execution (Damri and Vidyarthi 2016). However Dynamic analysis is performed in a secure isolated environment and the malware may behave differently in the two environments (virtual and real), in addition some actions of malwares are triggered under some specific conditions like (data, time, platform) in real environment which may not be detected by the secure virtual environment. Although dynamic analysis is essential complement to static analysis approach as it is very much preventive against code obfuscations.

2.4 Memory Analysis

Examining memory to spot concealed process is an effective detection method. For a malicious program to target machine it must load into memory, hackers uses sophisticated concealing techniques to hide their malicious programs, whereas every action adversaries take leave's a mark , malwares continues to progress as cyber criminals improves their techniques , For example malware deployed on a disk can be easily find out by forensic investigators therefore hackers shifted their tactics to host malicious software in a device Random Access Memory (RAM) , generally malwares added into RAM is not persistent meaning when a machine reboots the malware goes away but in today's technology world (Aghaeikheirabady et al. 2014) essential machines rarely reboot's, Hackers knows that and are exploiting the opportunity of non-persistent malwares. The amount of data traverse in RAM is huge and it makes it difficult for forensic analyst's to figure out inconsistences but the power of forensic tools are that they don't necessarily needs to looks for particularly type of attack (Hua and Zhang 2015) , all what forensic examiners check whether the operating system (OS) and all other related software's are running normally as it should. However, memory forensics alone is not sufficient to detect the changing behavior of malwares, this technique can just quickly assist the investigator to isolate the inconsistent behavior to a specific machine that are affected.

2.5 Automated Malware Analysis

Sandboxing system is used for automated malware analysis , Investigating malwares requires a lot of efforts and skills for security analyst, Studying malwares using sandbox makes it very easy to experiment and play around with suspicious files in a protected environment , sandbox is a security platform used often in automated analysis for executing untested programs or codes, Sandboxing platforms will quickly get some detailed information outlining what such suspicious file is when executed inside an isolated environment (Vasilescu et al. 2014), One of the known open source sandbox systems available is called Cuckoo; which dissect malwares in virtualized safe environment in a fast and effective manner. However, because of the polymorphic and metamorphic behaviors of the malwares which often change its code and features upon execution behaves differently for different platforms, date, time and technologies; therefore, investigating malwares behavior in isolated platforms might not be always same as in real scenarios.

2.6 Honeypot's for Malware Detection

Honeypot contains the resource's to attract the attackers and captures their methodologies, techniques and tools without attackers knowledge (Saxena et al. 2015) , The data gathered will help malware investigators to help learn different ways of attack types and hackers intention, high interaction honeypots are advanced honeypot solutions which are deployed in real operating systems(OS) , One of the example of high interaction honeypots is "honeynet" which is two or more honeypots that works together in deceiving or trapping the attacker (Haltaş et al. 2014). However along with great things honeypot does in misleading hackers it could also possibly risk other systems if attackers misuse its resources to launch attack on other devices, they are time consuming and it is hard to manage them in case of high interaction honeypots (Osorio et al. 2015).

3. HYBRID MODEL

A lot of the work done for malware detection is either on static or dynamic analysis, in order to get better result's, the hybrid model uses features from both static and dynamic analysis for malware detection, both analysis are performed on malicious and benign files dataset, Printable string information (PSI) features are extracted using static analysis and dynamic analysis fetches API system call sequences while in execution.

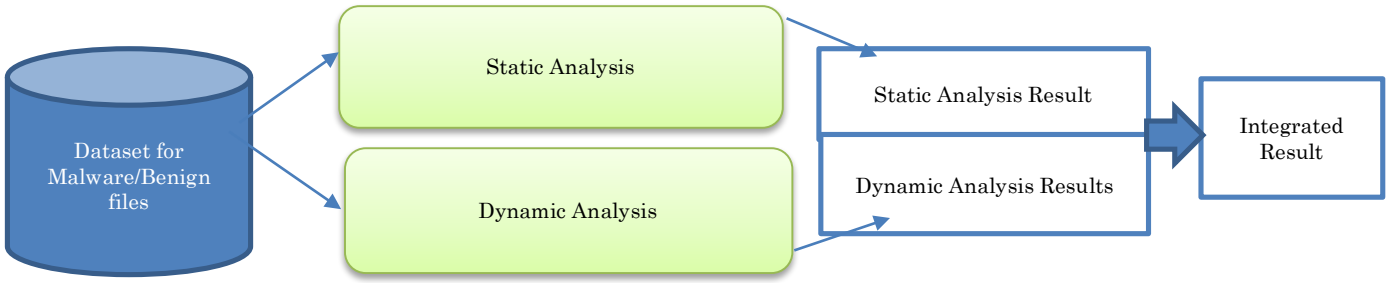

Fig. 1.

The printable string information (PSI) which is obtained from binary files are used as static features, un-encoded printable strings information are extracted from binary executable files , hackers using obfuscation techniques adds unwanted string files to mislead investigator therefore not all PSI strings fetched from binary files are useful , The extracted PSI are sorted and categorized according to the number of occurrence within a file and PSI's below define threshold are discarded. A feature list is created which will have all strings chosen from executable files in the dataset both malware and benign. Each sample of the malware and benign files are matched with the feature list and based on its strings they are represented by true/false binary value. Considering dataset three most effective files corresponding to three binary files are selected from global feature list which are "findFirstFile, "GetLongPathName" and "GetLastError" (Shijo and Salim 2015), this will be combined with the results from dynamic analysis technique which uses API system call sequences as shown in the figure 1 above.

Table I. Collaborative static and dynamic features dataset.

Class	FindFirstFile	GetLongPathName	GetLastError	3-Grams	4-Grams
Malware	1	1	0	1	0
Malware	1	0	1	0	1
Benign	0	0	1	0	0

Source: (Shijo and Salim 2015)

Dynamic analysis is performed for acquiring the API calls made by the binary file during execution , This experiment uses Cuckoo malware analyzer installed under Ubuntu on VMware virtual machine , The log files from the experiment contains list of all API call details while in execution such as registry modification , heap memory address and process address , The attacker misuses same set of API which operating system uses to access the low level hardware through system calls for the programs (Uppal et al. 2014) , Call sequence will be higher in the same class as compare to files in different classes , n-gram based method is used to analyze the call sequence called API-callgrams, Three and four API call grams are generated for each file from the processed call sequence log , n-grams below threshold is eliminated and the selected API-call-grams is considered as features, The table above shows the frequency of occurrence for 3-grams and 4-grams API-call-sequences , using both static and dynamic analysis if a file contains true for PSI strings and true in one of the API-call-grams it indicates a malware (Shijo and Salim 2015).

To reduce memory and processors utilization in malware detection, a new study approach (Milosevic et al. 2016) which captures the file properties, import functions and system call sequencing and based on these features identify and detect malicious files, Such as , if one file is known as malicious with clear characteristics it can then relate similar files with like features as also malicious. This technique to categorize the files into good and bad would prove to be efficient as a large dataset is not required to make malware classification, Therefore, observing metrics and behavior by extracting both static and dynamic features properties and categorizing it as good or bad would make the algorithm work efficient for the next suspicious files having same behavior and therefore it may lead to promising detection results. The ability to merge several malware traits together and comparing those findings to already defined good or bad files can provide better and more timely malware detection.

4. CONCLUSIONS

In this work using integrated model has proven that using both static and dynamic features together will increase the detection accuracy than standalone static or dynamic methods. As a solution, I propose a detection framework based on the PSI string and API system call sequences but it could be further enhanced by integrating machine learning and data mining as a feature set for detection in the existing solution.

REFERENCES

Aghaeikheirabady, M., Farshchi, S. M. R. and Shirazi, H. (2014) A new approach to malware detection by comparative analysis of data structures in a memory image. *2014 International Congress on Technology, Communication and Knowledge (ICTCK).* 26-27 Nov. 2014.

Damri, G. and Vidyarthi, D. (2016) Automatic dynamic malware analysis techniques for Linux environment. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom).* 16-18 March 2016.

Haltaş, F., Uzun, E., Şişeci, N., Poşul, A. and Emre, B. (2014) An automated bot detection system through honeypots for large-scale. *2014 6th International Conference On Cyber Conflict (CyCon 2014).* 3-6 June 2014.

Hua, Q. and Zhang, Y. (2015) Detecting Malware and Rootkit via Memory Forensics. *2015 International Conference on Computer Science and Mechanical Automation (CSMA).* 23-25 Oct. 2015.

Milosevic, J., Ferrante, A. and Malek, M. (2016) What does the memory say? Towards the most indicative features for efficient malware detection. *2016 13th IEEE*

Annual Consumer Communications & Networking Conference (CCNC). 9-12 Jan. 2016.

Moser, A., Kruegel, C. and Kirda, E. (2007) Limits of Static Analysis for Malware Detection. *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007).* 10-14 Dec. 2007.

Osorio, F. C. C., Qiu, H. and Arrott, A. (2015) Segmented sandboxing - A novel approach to Malware polymorphism detection. *2015 10th International Conference on Malicious and Unwanted Software (MALWARE).* 20-22 Oct. 2015.

Saxena, U., Bachhan, O. P. and Majumdar, R. (2015) Static and dynamic malware behavioral analysis based on arm based board. *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom).* 11-13 March 2015.

Shijo, P. V. and Salim, A. (2015) Integrated Static and Dynamic Analysis for Malware Detection. *Procedia Computer Science* 46, 804-811.

Uppal, D., Sinha, R., Mehra, V. and Jain, V. (2014) Malware detection and classification based on extraction of API sequences. *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* 24-27 Sept. 2014.

Vasilescu, M., Gheorghe, L. and Tapus, N. (2014) Practical malware analysis based on sandboxing. *2014 RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference.* 11-13 Sept. 2014.