Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Sophisticated Approach toward Recognition of Malicious Software

[1]Ashwin Kharat, [2]Ranveer Kumar, [3]Subhajit Chakraborty, [4]Prof J.E.Nalawade
[1, 2. 3] Computer Engineering, SPPU, India
[4]Asst. Professor, SIT, India

Abstract— In today's world there is immense growth of malicious software's, which results in lot of strain on computer world. So there is a need of accurate and efficient method for malicious software detection. Presently widely used methods for detection are anomaly and signature based technique. These methods are effective when either existing samples are present or malicious software signatures are known in databases. The proposed system is also signature based but the implementation is much more proficient and complexity is reduced significantly then previously existing systems. Our system along with existing technologies also scans each code present in the projects hence giving higher accuracy and less ramification then previous work.

Keywords— Malicious Software, Proficient, Detection, Ramification, Signature Based

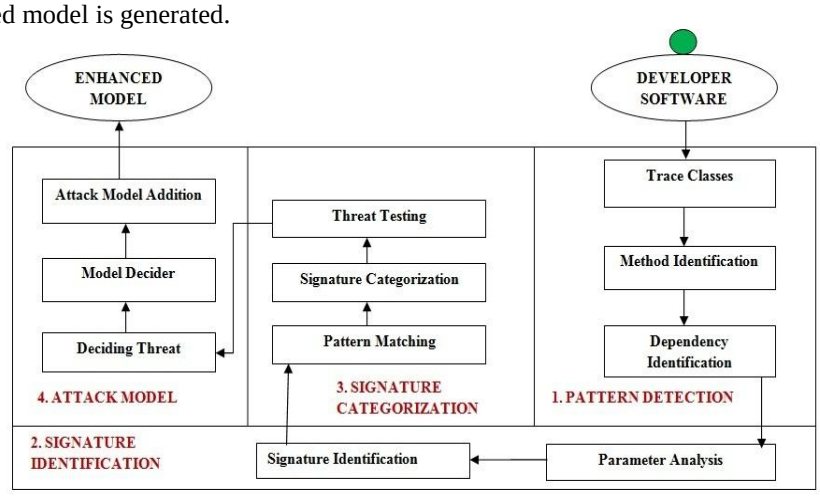
I. INTRODUCTION

In this project we use signature based technique to handle malicious software. Signature based technique provide significantly better result in catching the malicious software as it uses the inspection method on suspicions software. Signature based technique uses isolation method to isolate suspicious software. It creates a virtual environment in which the uncertified applications or software's are tested to the utmost level up to the code level in order to identify the fault. Here with the behavioural model we also test the code part for determining whether there in any fault in code section as well. Here uncertified application tested and initially we test its behaviour. After the behaviour is checked, we check the software code by code for any unusualness. Code scrutiny facilitates us to catch any hidden malware more precisely .If present, then the publisher take appropriate action against it and if not, then the application is allowed for user to install. At last, both the code and behavioural aspects are checked in order to determine the validity and configuration of the uploaded project to make it safer and error free.

Our project comprises of four sub models-

1. Pattern detection- Our system initially traces the classes following the identification of method and dependencies.
2. Signature identification- Then these elements are analysed along with signature recognition.
3. Signature Categorisation- Here the detected pattern are matched and then categorised signature are sent for threat testing.
4. Attack model- Here the detected threat are analysed, model is being generated and passed on for getting updated model.

First model detect the blueprint of the software or the project uploaded by tracing files in it. After detecting model project is sent for the cross substantiation where each details are analysed . Then we categorise it by verifying it for any abnormality detection. Then at last we prepare our assault model which is passed on for checking the mal functionality And hence restructured model is generated.



II. LITERATURE SURVEY

Malware stands for malicious software. Any software which tries to manipulate system or gather vital information is considered as malware. Malicious software [1] can easily be classified as

Virus- Its software that has harmful intention and it can duplicate itself one after another. Virus is always hidden in executable file or code and when the file is run it attack the system and starts to replicate.

Worms- Its duplicating malicious software which needs a network to run itself and it consume large amount of bandwidth. Worm is not like virus it does not need executable file it can directly attack the system.

Spyware- A spyware is malicious software that monitors and collects personal information of the user. Then it forwarded that information to the spyware created. It can enter the system when a certain free or trail software is installed.

Adware- Adware is all so called as ads supporting malicious software. It plays unnecessary ads to the user and hangs the system. Adware enter the system through free and trail games.

Trojan - Its replicate the behaviour of an authentication of any login it also access the vital information. It can also access user's personal information and it can damage importance resources.

Botnet - It's malicious software that remotely controls the target system. It is mostly used on public network. It can enter the system by any source and when it any command it make changes into the targeted machine.

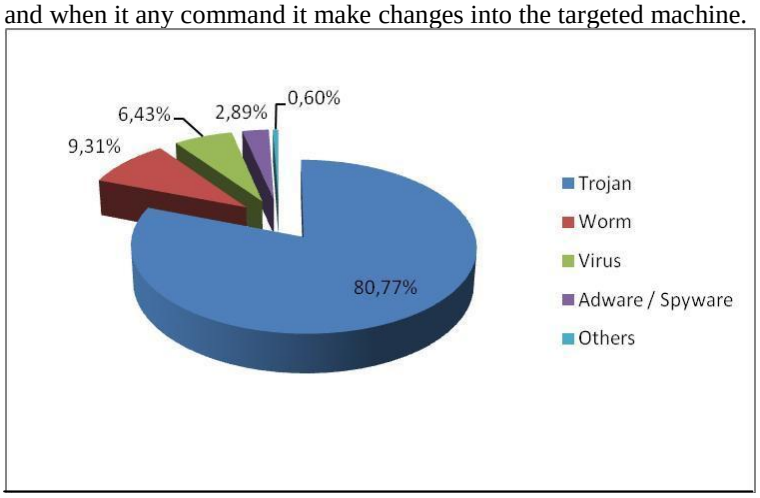
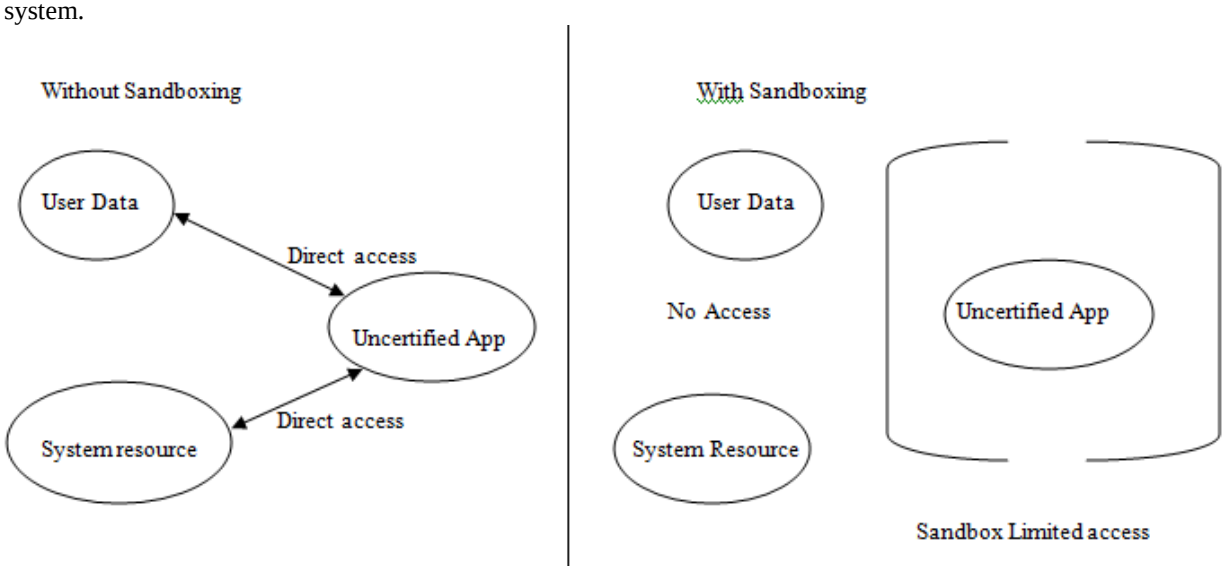


Fig1. Percentage attack of different malwares in user systems.

There are two widely used techniques for detection of malware. Anomaly and Signature based technique. Anomaly based technique is also called as heuristic technique it detect malicious software on pervious malware recognition it cannot detect newly evolve malware. Signature based technique detect malicious software by it signature , every time a new software is install it or run it check its signature and behaviour. Heuristic technique always needs new updates of new malicious software and its attack. It cannot handle newer malicious software which has not been detected previously. Signature based technique also keep database of attack and type of malicious software. But for every new uncertified program it checks its signature and test the program for any malicious behaviour.

Sandboxing technique [2] is a traditional technique of handling malicious software. Sandboxing works on isolation mechanism it limits any resource access to the malicious software.

Advantage of sandboxing is that uncertified program or application is isolated hence if it contains malicious properties then it won't harm system directly. Limitation to sandboxing is that it offers only one-way protection. Another issue is that sandboxing need some system exports and vital system files and if sandboxing is infected then it can harm the system.



Semantics-aware malicious software [3] detection resists more to frequent obfuscations used by malicious software. This detection uses decryption loops. It has limitation that it only detects malicious software which represents same ordering of memory allocation. It cannot handle malware which reorder its memory.

Malware transformer [4] it undoes the effect of the malware. Malware transformation can undo three techniques

1. Packing.
2. Code reordering.
3. Junk insertion.

Malicious software can damage a lot of vital data and to reverse the process we need malware transformer.

Behavioural prototype [5] identifies deed performed by malware rather than imitation markers. There is a clear difference among simulation-based authentication and formal authentication which are directly associated to the dynamic and static modes.

III. CONCLUSION

Thus, in this review paper we make use of a virtual environment where we test the untrusted application of developer before it can cause any fault by using pattern detection, signature justification and categorisation along with its derived attack model. Due to it possible errors can be detected well in advance and can be corrected according to the requirement finally giving an updated valid and error free software free from any vulnerabilities.

REFERENCES

[1] Vinod P and V.Laxmi and M.S.Gaur, *Survey on Malware Detection*, Department of Computer, Malaviya National Institute of Technology, Jaipur, Rajasthan.

[2] Manigandan Radhakrishnan and Jon A. Solworth, *Quarantining Untusted Entities: Dynamic Sandboxing using LEAP*, University of Illinois at Chicago,

[3] Mihai Christorescu and Somesh Jha, *Semantics-Aware Malware Detection*, University of Wisconsin, Madison:Proceedings of Conference Detection of intrusions and malware and vulnerability, Assessment, pp.64-87 (2008).

[4] Mihai Christorescu and Somesh Jha and Johannes Kinder and Stefan Katzenbeisser and Hwlmut Veith, *Software transformations to improve malware detection*:Springer-Verlag France 2007.

[5] Meng Zhang and Anand Raghunathan and Niraj Jha, *A defence framework againt malware and vulnerability exploits*:Springer-Verlag Berlin Heidelberg 2014.

[6] (2010) Kasperskey website. Available http://www.kaspersky.com

