Volume 5, Issue 6
December 2016

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Classifying Packed Programs as Malicious Software Detected

Edgar O. Osaghae

Department of Computer Science, Federal University Lokoja, Lokoja, Kogi State, Nigeria.

E-mail: edgarosaghae@gmail.com

ABSTRACT

The difficulty in revealing packed malware by malware analysts is slowing down malware analysis process however, the debate of whether to classify packed programs as malware is a decision antivirus community is yet to take. Currently, malware analysts use static and dynamic analysis techniques to unpack programs. The static analysis technique can help an analyst study the entire contents of the unpacked, but this technique requires expertise in reverse engineering skills to be used effectively. The dynamic analysis technique is easier to use because the suspicious packed programs are allowed to run inside an emulator, to reveal their unpacked contents. However, the entire process of revealing packed programs using dynamic analysis technique can be very slow and may not display the whole unpacked contents of the programs. Currently, some antivirus systems classify packed programs as malicious software, which the detection engine could not unpack. In this paper, an attempt was made to discuss four reasons why classifying packed programs as malware programs, is better than the difficulty involved in attempting to unpack a programs, prior to malware analysis. Firstly, it is claimed that more than 80% of the existing malware samples are packed, so, it is believed that most malware writers make used of packers. Secondly, a packed illegitimate program mistakenly classified as malware, should not be considered as a false alarm condition because, the purpose why it was packed was not defined to the user. Thirdly, if a forensic investigation is conducted on a user computer system and the found packed programs was classified and stolen from elsewhere, it could lead to a legal problem for the user. And lastly, the process of unpacking executable programs is very difficult and requires an expertise in reverse engineering. In conclusion, even if packed programs should be allowed to run in a computer system without classifying it as malware, a trusted third party should be involved to guarantee that the set of packed programs meant for a particular user, is malware-free.

Keywords: *Malware Analysis, Packer, Malware Scanners and Entropy Analysis*

1. INTRODUCTION

A Packer is an executable file that compresses another executable file for two main reasons, either to reduce the size of the executable file or to avoid being subjected to an analysis. Packing technique is a common obfuscation technique used by malware writers and this makes it easier for malware to hide their malicious codes from malware scanners. When a malware program has been packed, a malware analyst can have access to the packed malicious program and finds a way of unpacking it [3, 6]. The word *malware* means, malicious software. Malware damages other legitimate software programs in the computer system by disrupting their normal working process. There are two types of malware detection techniques, they are static and dynamic analysis. Static analysis is used to reveal the contents of malware by unpacking and disassembling the malware codes. While dynamic analysis is also used to reveal the contents of malware by observing their malicious behaviors, the malware is made to run inside an emulator. The most important defense method against malware, is to use the malware scanner. The common technique used by a malware scanner to detect a malware, is the use of malware signatures. The main functionality of a malware scanner depends on a regularly updated of their signatures database, and these signatures help the scanner to identify known malware [1, 2, 18, 9, 8].

The advent of malware has created an enormous challenge to the security of computers and computer controlled systems. Symantec Corporation in the year 2007, reported that a total of 499,811new malware samples were collected. Another antivirus company called F-secure, also reported that the total number of malware recorded in the year 2007 alone, is equivalent to the total number of malware produced in over a period of twenty years [2].

Developers of packers have different motives for writing their packers and this has led to arrays of many packers, with different routines [3]. Due to the differences in the routines inherent in packers, malware analysts have difficulties attempting to unpack malware codes, because malware writers prefer using custom-made packers to pack their malware codes. Currently, about 80% of malware are packed with numerous custom-made packers and this has created difficulty in unpacking malware. Due to the exponential increase in the number of malware in the wild, it is pertinent to find a way to justify that a detected packed program, should be classified as malware. In this paper, an attempt is made to justify an alternative approach to classify detected packed programs as malicious software.

2. LITERATURE REVIEW

Although packing is legitimately used for software protection such as, file compression but majority of malware programs, make use of packing. In one month during 2007, 79% of identified malware was packed. Additionally, almost 50% of new malware in 2006 were repacked versions of existing malware programs. Unpacking process is the only technique used by static analysis to reveal the hidden characteristics of malware. In the problem scope of unpacking, it can be seen that many instances of malware, utilizes identical or similar packers.

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Being able to automatically unpack malware, provides benefit in revealing the malware's real content and the behaviour relating to how the malware interacts with the computer system is seen. Emulation provides a safe and isolated environment for malware analysis. [8]. When a packed executable program is executed, then the unpacking routine in the packed executable program unpacks. Then the executable program unpacks its contents onto the process memory of the computer system. And it executes the unpacked code from the OEP (Original Entry Point). Examples of widely used packer are UPX, ASPack and Themida [5, 4, 11]. Executable unpacking analysis is based on: entropy analysis, use of identifiers and entropy analysis, improved entropy analysis and entropy related analysis. Entropy reduction techniques are used to improved entropy analysis [20, 10, 19].

The study presented by Symantec Research Laboratories showed, that over 80% of malwares is packed. Nevertheless, antivirus scanners still respond to packed malwares by updating signatures to detect newly packed malwares. The time to analyze packed malwares and find their signature, takes a long time compared to the time to create new packers. Consequently, some antivirus systems simply detect packed executable programs whose packers are the same as malware programs, leading to false alarms [17, 16, 15].

3. RELATED WORK

The challenges of revealing packed malware by malware researchers is huge however, researchers have attempted to describe the best ways to manage the unpacking process of a packed malware. Another challenge is the best to categorized a packed program during static analysis and determine whether to classify it as malware or subject to further analysis.

Robert and James [14] developed a tool called Bintropy which uses entropy analysis to help in analyzing and generate statistics on malware collections that contained packed or encrypted samples. The tool was good at identifying encrypted sections of the samples and at the same time, providing statistical data on large-sized malware sample collections at a low level of detail. The advantage of using the Bintropy is that, it can identify packed and encrypted executables. The authors noted that Bintropy is not a more fine-grained method in detecting packed malware samples. The authors recommended an improvement for Bintropy, by employing computation beyond simple frequency counting [14].

Sun [13] in his PhD thesis investigated the common features of packers and presented a novel, fast yet effective packer analysis framework called REFORM (Reverse Engineering For Obfuscation ReMoval). REFORM provides an extremely effective packer classification model based on a set of randomness measurements generated from a packed executable program. REFORM has the following advantages:

- Makes use of various statistical classifiers to achieve even better classification performance.
- Enables an efficient generic unpacking strategy that uses an ordered address execution histogram, to capture the memory after the unpacking loop has executed.
- It has the capability on speeding up packer detection, identification and unpacking procedures.

The drawbacks of REFORM are:

- It was not optimized to resist various armoring techniques, such as various anti-unpacking tricks applied by the packer and malware.
- There is no robust and scalable packer analysis system to keep up with the accelerating growth in packed malware [13].

According to [12], they proposed and evaluated the use of entropy analysis to identify variants of malware. The merits of the proposed entropy analysis are:

- It can effectively identify variants of malware in samples of real malware.
- The proposed system can demonstrate a high probability that new malware is a variant of existing malware.
- It is efficient in unpacking and classifying malware variants [12].

4. DISCUSSION

Enormous efforts have been made by existing malware researchers, on the most efficient way to unpack a malware. In this section, there is an attempt to discuss the justification while classifying packed programs as malware, is better.

- **Packed Program Issue:** It is claimed that more than 80% of the existing malware samples are packed. In other words, only about 20% or less of malware are not packed. Since antivirus detection systems are not perfect in detecting and eliminating malware (detection is not 100%), then detecting a packed program as malware will save antivirus communities a lot of time. It may be surprising to know that some existing antivirus systems detection rate may not be up to 80%. So, if their detection system is enhanced to detect packed samples as malware, their overall detection rate may improve tremendously.
- **False Alarms Issue:** When a packed illegitimate program is mistakenly classified as malware, it should not be considered as a false alarm condition in this instance, because the purpose why it was packed was not defined to the computer users. So, it is correct to assert that when illegitimate packed and non-malware programs are mistakenly classified as malware programs, the false alarm signals should be ignored.
- **Legal Issue:** An unwanted program that finds its way to a computer system, which is not useful to the users should be classified as malware, for legal reasons. Sometimes, some unwanted packed programs that mistakenly finds its way into another computer, may have been stolen or routed to a wrong destination. In fact, the user should be glad that such packed programs are discarded from his/her computer system.

If a forensic investigation is carried out on a user's computer and packed programs that are highly classified are found, the user may be held responsible for being in the possession of such programs.

- **Complexities in unpacking Process:** Before a malware is analyzed, firstly, it will be checked if it is packed or not. To compound the problem, unpacking an executable program statically is more challenging than a dynamic process. In most cases, the techniques used to unpack a program require advanced reversed engineering skills. Even if an expert malware analyst has expertise in unpacking malware programs, the process can delay malware analysis process. The point is that, if the owner of the packed program has genuine intention, why hiding the contents using a packer. Classifying packed programs as malicious software is the right is thing to do, because the purposes for packing the program in the first instance, was not defined to the computer user. In addition, it frees the computer system memory of unwanted programs hence, the computer performance will be greatly improved.

5. CONCLUSION

This paper discusses the reasons while it is better to classify packed programs as malware programs. The reasons are attributed to challenges faced, when a malware analyst is attempting to unpacked malware programs. It is observed that classifying packed programs as malware infected, would have little negative effects on the computer system and its contents. Firstly, it is claimed that more than 80% of the existing malware samples are packed, so, it is believed that most malware writers make used of packers, hence, it is better to classify packed program as malware. Secondly, a packed illegitimate program mistakenly classified as malware, should not be considered as a false alarm condition, because the purpose why it was packed was not defined. Thirdly, if a forensic investigation is conducted on a user computer system and a packed classified program is seen, it could lead to a legal problem, for the computer owner. And lastly, the process of unpacking programs is very complex and requires an expertise in reverse engineering. Even if the malware analyst is able to unpack a program, the process may slow down the overall malware analysis process.

In future, there will be an attempt to discuss how a third party member can mediate between the computer users and the owners of the packed program. This mediation will guarantee that a packed program about to be executed in the user's computer, is malware free. Considerations will be given to how the third party member will authenticate the prospective packed programs and ensure that it is malware-free. When the programs eventually get to the users computers and contains malware samples, the third party will take responsibility for the mistake.

REFERENCES

[1] J. H. Yang and Y Ryu, "Toward an Efficient PE-Malware Detection Tool", Advanced Science and Technology Letters, Vol. 109, pp. 14-17, 2015.

[2] J. S. Suganya, D. Krishnamoorthy and S. hiruniraiSenthil, "Analyzing and Detection of New Malware in Network Security", International Journal of Advanced Scientific and Technical Research, Vol. 6, No. 2, pp. 607-612, 2013.

[3] D. Devi and S. Nandi, "PE Features in Detection of Packed Executables", International Journal of Computer Theory and Engineering, Vol. 4, No. 3, pp. 476-478, 2012.

[4] O. A. Adesegun, "A Review of the Effectiveness of Malware Signature Databases against Metamorphic Malwares", Control Theory and Informatics, Vol. 5, No. 2, pp. 18-19, 2015.

[5] D. Shin, C. Im, H. Jeong, S. Kim and D. Won, "The New Signature Generation Method Based on an Unpacking Algorithm and Procedure for a Packer Detection", International Journal of Advanced Science and Technology, Vol. 29, pp. 83-100, 2011.

[6] A. Devi and G. Aggarwal, "Manual Unpacking of UPX Packed Executabl Using Ollydbg and Importrec", IOSR Journal of Computer Engineering, Vol 16, No. 1, pp. 71-77, 2014.

[7] M. M. K. Al-Anezi, "Generic Packing Detection using Several Complexity Analysis for Accurate Malware Detection", International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, pp. 7-14, 2014.

[8] V. P. Vandana, "Flowgraph Based Classification System for Packed and Polymorphic Malware", International Journal of Research in Information Technology, Vol. 2, No. 7, pp. 69-73, 2014.

[9] A. M. Al-Bakri, H. L. Hussein, "Static Analysis Based Behavioural API for Malware Detection using Markov Chain", Computer Engineering and Intelligent Systems, Vol. 5, No. 12, pp. 55-63, 2014.

[10] Silvio C., Yang Xiang, "Classification of Malware Using Structured Control Flow", Proceedings of 8th Australasian Symposium on Parallel and Distributed Computing, Brisbane, Australia, pp. 61-70. 2010

[11] K. Kaushal, P. Swadas, N. Prajapati, "Metamorphic Malware Detection Using Statistical Analysis", International Journal of Software Computing and Engineering, Vol. 2, No. 3, pp. 49-53, 2012.

[12] M. R. Pasha, Y. Prathima and L. Thirupati,

"Malware System for Packed and Polymorphic Malware", International Journal of Advanced Trends in Computer Science and Engineering, Vol 3, No. 1, pp. 167-172, 2014.

[13] L. Sun, " REFORM: A Framework for Malware Packer Analysis Using Information Theory and Statistical Methods", PhD Thesis RMIT University, 2010.

[14] L. Robert, H. M. James, "Using Entropy Analysis to Find Encrypted and Packed Malware", IEEE Computer Society, pp. 40-45, 2007.

[15] M. Z. Shafiq, S. M. Tabish and M. Farooq, "PE-Probe: Leveraging Packed Detection and Structural Information to Detect Malicious Portable Executables, Next Generatio Intelligent Networks Research Center", National University of Computer and Emerging Sciences, Islamabad, Pakistan, pp. 1-10, 2009.

[16] Y. Choi, I. Kim, J. Oh and J. Ryou, "Encoded Executable File Detection Technique via Executable File Header Analysis", International Journal of Hybrid Information Technology, Vol. 2, No. 2, pp. 25-36, 2009.

[17] G. Jeong, E. Choo, J. Lee, M. Bat-Erdene and H. Lee, "Generic Unpacking using Entropy Analysis", IEEE Journal pp. 114-121, 2010.

[18]. G. M. W. Al-Saadoon and H. M. Y. Al-Bayatti, "A Comparison of Trojan Virus Behaviour in Linux and Windows Operating Systems", World of Computer Science and Information Technology Journal, Vol. 1, No. 3, pp. 56-62, 2011.

[19] S. R. White, "Open Problems in Computer Virus Research", Virus Bulletin Conference, Munich, Germany, pp. 1-11, 1998.

[20] E. O. Osaghae, "Packed Malware Detection using Entropy Related Analysis: A Survey", Journal of Engineering, Vol. 5, No. 11, pp. 59-61, 2015.

AUTHOR PROFILES

Edgar O. Osaghae received his BSc, MSc and PhD degrees in Computer Science from University of Benin, Benin City, Nigeria, in 2001, 2005 and 2013 respectively. His research is focused on Computer Virology, Compiler Constructions, Computer Algorithms and Software Engineering. Currently, he is a Senior Lecturer at Department of Computer Science, Federal University Lokoja, Kogi State, Nigeria.