IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 05, Issue 11 (November. 2015), ||V2|| PP 59-61

Packed Malware Detection using Entropy Related Analysis: A Survey

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

Abstract: - The challenges of revealing packed malware by security analysts is enormous however, malware researchers have tried to use entropy analysis to detect packed malware. Since the problem of revealing packed malware is the time taken to unpack a malware and the analysis which helps to reveal the packed malware is centered on entropy analysis. However, the need arises to determine if a packed malware is revealed using entropy analysis and if it is not, how is the analysis techniques related to entropy analysis. Consequently, a survey was conducted on packed malware research works, using entropy and entropy related analysis. The survey was categorized into two tiers namely, executable unpacking analysis and unpacking identification. Executable unpacking analysis shows that the packed malware is based on: entropy analysis, use of identifiers and entropy analysis, improved entropy analysis and entropy related analysis. With the survey on unpacking identification, shows that pattern recognition techniques and computed confidence interval, make use of entropy analysis. Original entry point, the use of UPX and emulations techniques, are used as identifiers with entropy analysis to detect packed malware. Entropy reduction techniques are used to improved entropy analysis. Attributes in executable file header and complexity measured by several algorithms are used as entropy related analysis. The survey shows that research works on packed malware detection, either make use of entropy or entropy related analysis.

Keywords: *Malware Detection, packer, Entropy Analysis*

I. INTRODUCTION

Malware is a generic term used to describe all kinds of malicious software like Viruses, Worms, Spywares and Trojan horses. It is created by computer attackers to cause major threat to the security and privacy of computer users, which may be responsible for a significant amount of financial loss [10], [11], [13], [14] and [16]. The number, power and variety of malware programs increase every year, as does their ability to overcome all kinds of security barriers. Current commercial anti-malware solutions rely on a signature database. A signature is a sequence of bytes that is always present within a malicious executable. However, malware writers use code obfuscation techniques to hide the actual behaviour of their malicious creations. Executable packing is one of the most popular obfuscation techniques, among all obfuscation techniques. Packers are software programs that compress and encrypt other executable files in a disk and restore the original executable images when the packed files are loaded into memories. An example of packer software is Ultimate Packer for eXecutables (UPX). UPX was released in March 1998 and it is one of the most popular packer software used world-wide [5], [1], [2], [9], [12], [15] and [17].

Entropy is a method for measuring uncertainty in a series of numbers or bytes. In technical terms, entropy measures the level of difficulty or the probability of independently predicting each number in the series [5] and [18]. Entropy can be considered as one of the major feature in classification of packed and non-packed executables. That is why entropy of a packed executable is always higher than a non-packed executable [2]. A set of metrics are developed that analysts can use to generalize the entropy attributes of packed or encrypted executable and thus, distinguish them from native (non-packed or unencrypted) executables. Entropy analysis examines the statistical variation in malware executables, enabling analysts to identify packed and encrypted samples quickly and efficiently [1]. An example of an entropy tool is Bintropy. Bintropy is a prototype analysis tool that estimates the likelihood that a binary file contains compressed or encrypted bytes. Bintropy calculates other entropy-related file attributes, including the average and highest entropy scores. Bintropy helps analysts determine which executable sections might be encrypted or packed [5].

This paper presents a survey of packed malware detection techniques using entropy related analysis. The techniques are categorized based upon a two tier hierarchy that includes executable unpacking analysis and unpacking identification. The executable unpacking analysis reviews the entropy analysis, used of identifiers/entropy, improved entropy analysis and entropy related analysis. The unpacking identification section comprises of pattern recognition techniques, computed confidence intervals, original entry point, Untimate

Packer for eXecutable(UPX), emulation techniques, entropy reduction techniques, attributes in executable file header and complexity measured by several algorithms. This paper also provides the readers with information proclaiming that the solution to detecting a packed malware is either in entropy or entropy related analysis. This served as the main contribution of this paper.

II. SUMMARY

Among the set of research works on the use of entropy to detected packed malware in this research, the first found published work is [7]. [7] proposed the use of universal unpacker to detect and extract encrypted code from packed executables in a less-expensive computational way. They proposed a pattern recognition technique based on entropy of the executable packed file. Their main objective was to efficiently and accurately distinguish between packed and non-packed executables, so that only executables detected as packed will be sent to a universal unpacker, thus, saving a significant amount of processing time. This was followed by the pioneering work of [5], where they proposed the use of entropy and its management to identify packed or encrypted malware executables. They derived two entropy metrics based on the computed confidence intervals for average and highest entropy. If both the bin-trophy compared average file entropy and highest entropy block score exceed these respective values, it is labeled a malware executable as packed or encrypted.

The second step in this survey is the detection of packed malware by using identifiers. In [3], they proposed a generic unpacking mechanism to find the Original Entry Point(OEP) using entropy analysis. [2] extracts various features of portable executables, analyses the extracted features and comes up with best set of features. These best set of features can be used to identify whether a given binary is packed or not by using a Ultimate Packer for eXecutable (UPX) packer. They claimed that their approach can make it easier to figure out whether an executable is packed or not by using UPX. [8] proposed an approach based on static and dynamic analysis to classify malware. They employed entropy analysis initially to determine if the binary has undergone a code packing transformation. If packed, dynamic analysis employing application level emulation reveals the hidden code using entropy analysis to detect when unpacking is complete. They claimed that the system can easily unpacked malware.

The third step in this survey is to review the improvement made on entropy analysis. [6] claimed that the used of entropy score alone to distinguish between packed and non-packed files is not enough, they reviewed some common techniques of entropy reduction, and presented an advance one. They further observed that entropy-based detection of packed and encrypted malware is not effective by formalizing an entropy reduction attack. They proposed an approach to identify obfuscated file based on anomalies in their control flow graph and other instruction-based characteristics. They used this technique to develop a detection system that combines these features with other file structural features and lead to a very good result of detecting obfuscated malware.

The fourth step in this survey is to review the detection of packed malware based on entropy related techniques. [1] proposed a detection techniques on packed file based on its Portable Executable (PE) Header Analysis. They observed that in many cases, to packed and unpacked an executable codes, the PE files have unusual attributes in their PE headers. A characteristics Vector (CV) are utilized to detect the packed files. They claimed that their unpacking approach can only check only about the PE file's header area, whether the file is packed or not and did not extend their research more than that point.

Table 1: Summary

Executable Unpacking Analysis	Unpacking Identification	Examples
Entropy Analysis	Pattern Recognition Techniques	[7]
	Computed Confidence Intervals	[5]
Use of Identifiers and Entropy Analysis	Original Entry Point (OEP)	[3]
	Ultimate Packer for eXecutable (UPX)	[2]
	Emulation Techniques	[8]
Improved Entropy Analysis	Entropy Reduction Techniques	[6]
Entropy Related Analysis	Attributes in Executable File Header	[1]
	Complexity measured by several algorithms	[4]

[4] also proposed a packed file detection technique based on complexity measured by several algorithms, and it has tested using a packed and unpacked dataset of file type .exe. They proposed that the detection technique has 96% detection rate on packed files and 93% detection rate on unpacked files. Their experiments also

demonstrated that this generic technique can effectively prepared to unknown, obfuscated malware and cannot be evaded by known evade techniques.

Table 1 summarizes and assigns each work to the categories used in this survey research.

REFERENCES

[1] Y. S. Choi, I. K. Kim, J. C. Ryou, Encoded Executable File Detection Technique via Executable File Header Analysis, *International Journal of Hybrid Information Technology*, 2(2), 2009, 25-36.

[2] D. Devi, S. Nandi, PE File Features in Detection of Packed Executables, *International Journal of Computer Theory and Engineering*, 4(3), 2012, 476-478.

[3] G. Jeong, E. Choo, J. Lee, M. Bat-Erden, H. Lee, Generic Unpacking using Entropy Analysis, *The IEEE journal*, 2010, 114-121.

[4] M. K. Al-Anezi (2014). Generic Packing Detection using Several Complexity Analysis for Accurate Malware Detection, *International Journal of Advanced Computer Science and Application*, 5(1), 2014, 7-14.

[5] L. Robert, H. James, Using Entropy Analysis to Find Encrypted and Packed Malware, *IEEE Computer Society*, 2009, 40-45.

[6] Saleh, E. P. Ratazzi, S. Xu, Instruction-Based Detection of Sophisticated Obfuscation and Packing, *Department of Electrical Engineering and Computer Science, Syracuse University, Texas, USA*, 2014.

[7] R. Perdisci, A. Lanzi, W. Lee, Classification of Packed Executables for Accurate Computer Virus Detection, Pattern Recognition Letters, 29(14), 2008, 1941-1946.

[8] R. M. Pasha, Y. Prathima, L. Thirupati, Malware System for Packed and Polymorphic Malware, *International Journal of Advance Trends in Computer Science and Engineering*, 3(1), 2014, 167-172.

[9] L, Durfina, J. Kroustek, Z. Petr, D. Kolar, T. Hruska, K. Masarik, A. Meduna, Design of a Retargetable Decompiler for a Static Platform-Independent Malware Analysis, *International Journal of Security and Its Applications*, 5(4), 2011, 91-105.

[10] A. Alatabbi, M. Al-Jamea, C. S. Illiopoulos C. S, Malware Detection using Computational Biology Tools, *International Journal of Engineering and Technology*, 5(2), 2013, 315-319.

[11] A. M. Ali, M. A Maarof, Enhancing Malware Detection using Innate Immunization, *International Journal of Computer Science and Network Security*, 13(10), 2013, pages 74-77.

[12] L. Santo, X. Ugarte-Pedrero, F. Brezo, P. G. Bringas, NOA: An Information Retrieval Based Malware Detection System, *Computing and Informatics*, 32, 2013, 1001-1030.

[13] M. Apel, C. Bockermann, M. Meier (2009). Measuring Similarity of Malware Behaviour, The 5th LCN workshop on security in Communication Networks, Zurich, Switzerland, pages 891-898.

[14] R. Veeramani, R Nitin, Windows API based Malware Detection and Framework Analysis, International Journal on Scientific and Engineering Research, 3(3), 2012, 1-6.

[15] W. Yan, Z. Zhang, N. Ansari N, Revealing packed Malware, *IEEE Security and Privacy*, 2007, 65-69.

[16] A. E. Elhadi, M. A. Maarof, B. A. Barry, Improving the Detection of malware Behaviour using Simplified Data Dependent API Call Graph, *International Journal of Security and Its Applications*, Vol. 7, No. 5, 2013, 29-42.

[17] D. Uppal, V. Mehra V. Verma, Basic Survey on Malware Analysis, Tools and Techniques, *International Journal on Computational Sciences and Applications*, Vol. 4(1), 2014, 103-112.

[18] A. M. Al-Bakri, H. L. Hussein, Static Analysis Based Behaviour API for Malware Detection using Markov Chain, Computer Engineering and Intelligent Systems, 5(12), 2014, 55-63.