ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

PREVENTING PERSONAL COMPUTER FROM POLYMORPHIC MALWARES

S.Ananthi [*], V.Shyamala

[*] Department of ECE, MNSK College of Engineering, Pudukottai, Tamilnadu, India
Department of ECE, MNSK College of Engineering, Pudukottai, Tamilnadu, India

ABSTRACT

The system security and control flows always falls into the major issue of detecting the malwares, which plays various important roles and proves the effect of it to anywhere of the personal computers, these category of malwares are usually called polymorphic malwares. The polymorphic malwares takes several forms to affect or interrupt the users or user activities into the pc, there are lots and lots of difference between normal virus programs and malwares. A normal virus program cause affection in computer parts and destroy the hardware, it can easily be identified by means of various software and the malwares present like a normal program and cause resulting damage in user tasks or interrupt them in worst case. This system proposes a new novel variant methodology to efficiently detect the malwares in different stages. The malwares are identified by the detector in two different stages like string based signature identification and pre-filtering methodology. A minimum matching distance method is used to identify the malwares by means of differentiate it into the regular programs created by the users for legal purposes. For all the proposed system proves that the detection of polymorphic malwares using novel methodology creates an illusion in the security mean and prove the effectiveness of false positive rate as well as this method results better than all the existing methodologies.

KEYWORDS: Polymorphic malwares, String based signature identification, Pre-filtering methodology, Novel methodology.

INTRODUCTION

Malicious software presents a significant challenge to modern desktop computing. According to the Symantec Internet Threat Report, 499,811 new malware samples were received in the second half of 2007. In 2010, over 1.5 billion malicious code detections were identified by the same vendor. F-Secure published, "As much malware was produced in 2007 as in the previous 20 years altogether". This trend is continuing and makes the detection of malware before it adversely affects computer systems highly desirable. To achieve this, static detection of malware is still the dominant technique to secure computer networks and systems against untrusted executable content. Detecting malware variants improves signature based detection methods.

The size of signature databases is growing exponentially, and detecting entire families of related malicious software can prevent the blowout in the number of stored malware signatures. Malware classification and detection can be divided into the tasks of detecting novel instances of malware, and detecting copies or variants of known malware. Both tasks require suitable feature extraction, but the class of features to be extracted is often dependant on which problem is trying to be solved. Detecting novel samples primarily uses statistical machine learning. On the contrary, malware variant detection uses the concept of similarity searching to query a database of known instances. These similarity queries or nearest neighbor searches are known in machine learning as instance-based learning. Instance-based learning uses distance functions to show dissimilarity and hence similarity between objects. If the distance function has the mathematical properties of a metric, then algorithms exist that enable more efficient searching than an exhaustive set of queries over the database.

Traditional and commercial malware detection systems have predominantly utilized static string signatures to query a database of known malware instances. Static string signatures capture sections of the malwares' raw file content that uniquely identifies them. String signatures have been employed because they have desirable performance characteristics that enable real-time use. However, string signatures perform poorly when faced with polymorphic malware variants. Exact string matching

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

also ineffectively handles closely related but non-identical signatures.

Polymorphic malware variants have the property that the byte level content of the malware changes between instances. This can be the result of source code modifications or self-mutation and obfuscation to the malware. Signatures that rely on fixed byte level content are unable to capture the invariant characteristics between these polymorphic instances. To extend the capabilities of string based signatures, code normalization has been investigated. Using code normalization, programs are transformed into an ideally canonical representation to eliminate the mutations and obfuscations incurred by polymorphic variants.

The normalized program is then analyzed using existing string based malware detection systems. Code normalization improves the effectiveness of string based scanning, but has not been widely adopted due to efficiency concerns. Efficient real-time systems have been proposed that examine the runtime behavior of programs to identify malicious behavior. Malicious behavior can either conform to a policy of malicious intent, or reassemble the behavior of a program instance, known in advance to be malicious. However, static detection of malware has advantages it does not require conditional, untrusted or sandboxed execution of malware once the original contents of the malware are visible. Unpacking is the processing of revealing that code and typically occurs before the malware performs its malicious intent.

Many antivirus products implement static unpacking for known packers, and this accounts for the majority of samples. However, for novel packing techniques unpacking is often a dynamic process making effective static analysis against novel malware a hybrid approach. Additionally, snapshots of process images can be taken at runtime, thus avoiding the most common packing issues and can be used to statically identify if those processes belong to a known malware family. A variety of algorithms have been employed to statically detect malware variants with superior classification compared to string based approaches. An n-gram is one of all possible fixed sized substring extracted from a larger string. Our work is directly related to the n-gram concept. N-grams of byte level, or instruction level content, utilizing machine learning and classification has been proposed. However, n-grams are ineffective with polymorphic malware because of the changes the instruction level content.

A Replacement to Traditional Antivirus

Traditional AV suffers from the inability to detect malware variants efficiently from large databases. Control flow is effective and our system makes such a system practically efficient when using large databases. Moreover, it would reduce the size of the database required on the end host due to requiring fewer samples to recognize a large malware family. Malware classification can help group samples together to create traditional AV signatures, and our work improves the performance of this, yet our work also presents a direction for a replacement of user-level AV to cope with today's malware problem.

To Cluster Interesting Samples

AV vendors need to know which malware families are significant enough that they require manual analysis. Our system could be used to identify variants and group them to their family. If many instances of a family are identified, then that family may require human analysis to determine what the real impact of the malware is. Moreover, interesting samples, such as state-sponsored malware, can be used as a query to retroactively find any other related malware from databases of unlabelled samples.

Incident Response

An accurate system that identifies what family of malware a sample belongs to could be used in incident response. If a sample is a polymorphic variant of known malware, but is as yet unidentified by AV, an analyst could identify the family it belongs and hence have insight into what disinfection procedures are required and what impact the malware has on an infected system.

RELATED WORK

The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

The Symantec Internet Security Threat [1] report provides an overview and analysis of Internet threat activity worldwide, a review of known vulnerabilities, and highlights of malicious code. Trends in phishing and spam are also assessed, as are observed activities on underground economy servers. Volume XV of the Symantec Global Internet Security Threat Report will alert readers to trends and impending threats that Symantec has observed for all of 2009. The report is based on data collected from millions of Internet sensors, first-hand research and active monitoring of hacker communications.

Scanning files for signatures is a proven technology, but exponential growth in unique malware programs has caused an explosion in signature database sizes. One solution to this problem is to use string signatures, each of which is a contiguous byte sequence that potentially can match many variants of a malware family. However, it is not clear how to automatically generate these string signatures with a sufficiently low false positive rate. Hancock is the first string signature generation system [2] that takes on this challenge on a large scale. To minimize the false positive rate, Hancock features a scalable model that estimates the occurrence probability of arbitrary byte sequences in good ware programs, a set of library code identification techniques, and diversity-based heuristics that ensure the contexts in which a signature is embedded in containing malware files are similar to one another. With these techniques combined, Hancock is able to automatically generate string signatures with a false positive rate below 0.1%.

Metamorphic malware changes the structure of its code from infection to infection. This makes it very hard to classify or to detect. While the byte-sequence of two variants may be completely different, the core functionality of the malware has to stay the same. This includes the use of flags and constants that have to be consistent at specific points. We present a novel approach that allows us to detect metamorphic variants. Based on this detection, it is also possible to classify new samples to a metamorphic family. Our approach [3] identifies variants by tracking the use of consistent values throughout the malware. Our evaluation shows a 100% detection rate with 0 false positives for all metamorphic samples that do not change their behavior.

Nearly every malware sample is sheathed in an executable protection which must be removed before static analyses can proceed. Existing research has studied automatically unpacking certain protections, but has not yet caught up with many modern techniques. Contrary to prior assumptions, protected programs do not always have the property that they are reverted to a fully unprotected state at some point during the course of their execution. This work provides a novel technique for circumventing one of the most problematic features of modern software protections, so-called virtualization obfuscation. The technique enables analysis of heretofore impenetrable malware [5].

Due to the rapid development of computer technology and new methods for the extraction of data in the last few years, more and more applications of databases have emerged, for which an efficient and effective similarity search is of great importance. Application areas of similarity search include multimedia; computer aided engineering, marketing, image processing and many more. Special interest adheres to the task of finding similar objects in large amounts of data having complex representations. For example, set-valued objects as well as tree or graph structured objects are among these complex object representations. The grouping of similar objects, the so-called clustering, is a fundamental analysis technique, which allows to search through extensive data sets. The goal of this dissertation is to develop new efficient and effective methods for similarity search in large quantities of complex objects [6]. Furthermore, the efficiency of existing density-based clustering algorithms is to be improved when applied to complex objects.

PROPOSED APPROACH

In proposed approach a novel methodology is introduced to detect the polymorphic malwares. A new control flow mechanism is defined to prevent the malwares, which is efficient even the size of the database is large. The novel approach repeatedly does the same task so the performance is not harder compare to the existing approach. The signature of the malwares is really identical like the normal programs so this system uses the string based signature verification in its first stage; it efficiently detects the polymorphic malwares.

- A new novel variant method is introduced, which gives more efficiency and effectiveness to find polymorphic malwares.
- Efficient during large databases
- Malware Programs signatures are similar to normal programs so it is very hard to differentiate them, this system solve this by means of NCD and BLAST algorithms.
- Low Cost is enough for solving the problem.
- Performance is higher than the regular anti-virus schemes.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

NCD Algorithm
Normalized compression distance is way of measuring the similarity between two objects, be it two documents, two letters, two emails, two music scores, two languages, two programs, two pictures, two systems, two genomes, to name a few. Such a measurement should not be application dependent or arbitrary. A reasonable definition for the similarity between two objects is how difficult it is to transform them into each other.

Blast Algorithm
BLAST is abbreviated as "Basic Local Alignment Search Tool", which compares the sequence of information one by one and produces the analytical results as well. Basically it is used in biological terms. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold. Different types of BLASTs are available according to the query sequences. For example, following the discovery of a previously unknown gene in the mouse, a scientist will typically perform a BLAST search of the human genome to see if humans carry a similar gene; BLAST will identify sequences in the human genome that resemble the mouse gene based on similarity of sequence.

Polynomial Time Algorithm
In computer science, the time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the size of the input to the problem. The time complexity of an algorithm is commonly expressed using big O notation, which suppresses multiplicative constants and lower order terms. When expressed this way, the time complexity is said to be described asymptotically, i.e., as the input size goes to infinity.
Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, where an elementary operation takes a fixed amount of time to perform. Thus the amount of time taken and the number of elementary operations performed by the algorithm differ by at most a constant factor. Since an algorithm may take a different amount of time even on inputs of the same size, the most commonly used measure of time complexity, the worst-case time complexity of an algorithm, denoted as T(n), is the maximum amount of time taken on any input of size n. Time complexities are classified by the nature of the function T(n). For instance, an algorithm with T(n) = O(n) is called a linear time algorithm, and an algorithm with T(n) = O(2n) is said to be an exponential time algorithm.

Pre-filtering Algorithm
Pre-filtering algorithm treat a source as an identity based object, and compute the signatures based on the overlap of the content into the objects with its complexity like extensions. These techniques compute the signature of a created object and maintain the signature into the database for future analysis.

Structuring Algorithm
Structuring is a reverse engineering and recompilation technique to transform a control flow graph into its high level source code representation. We use a structuring algorithm to transform the control flow graphs into strings. The perception is that similar control flow graphs are structured into similar strings.

Smith-Waterman Algorithm
The Smith-Waterman algorithm gives the optimal local sequence alignment between two strings. The local sequence alignment seeks to provide an alignment between two strings taking into account the alignment of substrings. Local sequence alignment is used often in the field of Bioinformatics to identify similarity between genome sequences. It forms a metric allowing for metric access methods for indexing and searching. The Smith-Waterman algorithm has quadratic runtime complexity like the Levenshtein distance. The Basic Local Assignment Search Tool (BLAST) approximates the Smith-Waterman algorithm using a heuristic search. BLAST is used frequently to improve the efficiency of genome searches. We propose using off-the-shelf BLAST software to perform similarity searches of our malware signatures. The BLAST algorithm does not employ distance metrics for the similarity search, but uses the notion of an expected value, which describes the statistical probability of the occurrence of a random signature.

SYSTEM DESIGN

System Architecture
The major part of the project development sector considers and fully survey all the required needs for developing the project. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

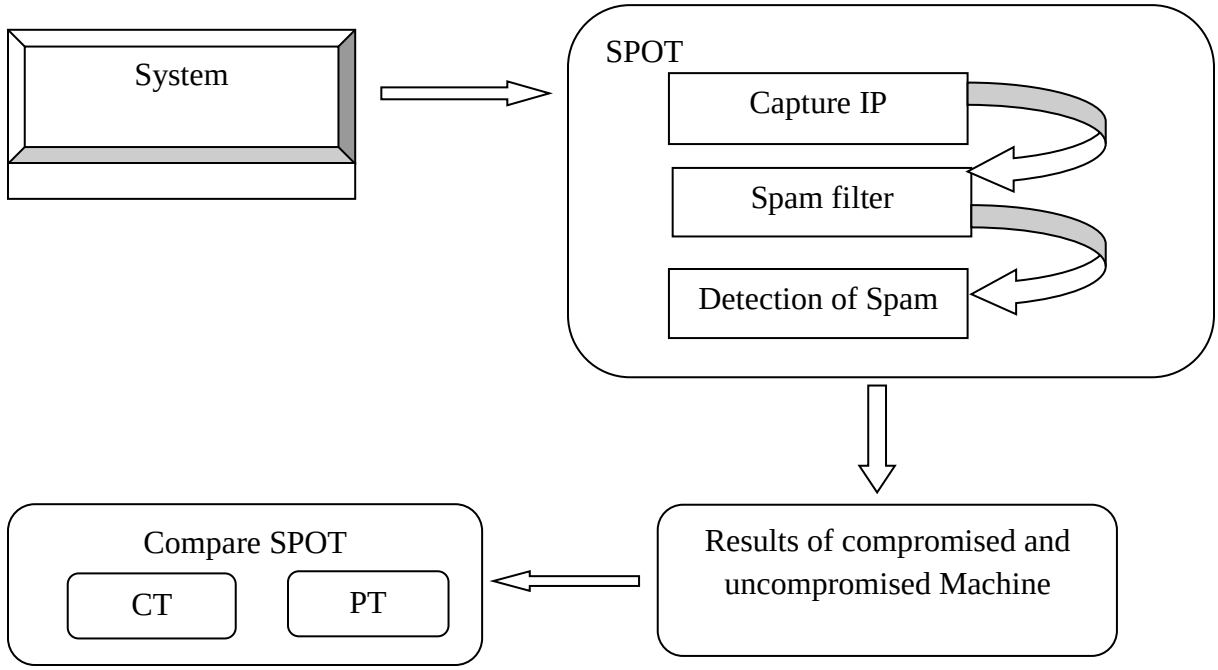

Fig 3: System Architecture

necessary software are needed to proceed with the next step such as developing the tools, and the associated operations. Generally algorithms shows a result for exploring a single thing that is either be a performance, or speed, or accuracy, and so on. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them.

Modules

1. Account authentication
2. Sending mails
3. SPOT detection
   a. SPOT Filter
   b. SPOT results
4. CT Detection
5. PT detection

Account Authentication

In this module the system is to check for the mail id and password is valid or not. If these two parameters are valid, then the system allows the user into their account and accesses all the features of the system. Otherwise is not valid.

Sending Mails

In this module, the system allows the user to send one or more mails to other user into the community. This mails either spam or non spam. Spam means the more copies of the single message are send. And it contains more than the administrator specified number of lines into the file.

SPOT Detection

In this module to capture the IP address of the system. That system mails are applied to filtering process. In this process, the mail content is filtered. Finally the system is to produce the result of filter.

CT Detection

In this module to set the threshold value Cs. Cs denotes the fixed length of spam mail. Also the process is to count the number of lines in each incoming mail. If the each mail, counts are greater than equal to threshold value. So, these mails are spam mail.

PT Detection

In this module the system is to set two threshold values. They are:

1) Ca- specifies the minimum number of mail that machine must send.

2) P- specifies the maximum spam mail percentage of a normal machine.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

This algorithm is used to compute the count of total mails and the count of spam mails of machine. To check this count of total mails are greater than equal to Cs and the count of spam mails are greater than equal to P. If it's true these mails are spam mail.

CONCLUSION

By using this new approach malwares can easily be classified from normal programs. In this approach a novel variant method is followed to identify the moderations and activities of the polymorphic malwares. The new system controls the activity of malwares and produces the results as much as in extreme manner. The NCD and BLAST algorithms are efficiently used to identify the signature of the malwares and control the affection of it. The number of false positives was low, and the efficiency of the prototype demonstrated that the system could be used on a desktop system and in future it will be used in e-mail gateway system.

FUTURE WORK

The control flow based malware variant detection scheme is implemented with PT and CT threshold value set approach, but in future a new module of SPOT detection is introduced, which is used to capture the IP address of the system and mails are applied to filtering process. In this process, the mail content is filtered. And also in this system needs to set the threshold value Cs. Cs denotes the fixed length of spam mail. Also the process is to count the number of lines in each incoming mail. If the each mail, counts are greater than equal to threshold value. So, these mails are spam mail. In future this CT and PT based malware checking is also need to be implemented, which helps to compute the count of total mails and the count of spam mails of machine. To check this count of total mails are greater than equal to Cs and the count of spam mails are greater than equal to P. If it's true these mails are spam mail. And also this malware detection system will be implemented in mobile based approach in future.

ACKNOWLEDGEMENTS

Our completion of this paper could not have been accomplished without the support of staff members those who are working with us. We are very much thankful to them. For the reference, we refer many articles and research papers of many authors and institutions those are available in online and offline. We offer our sincere appreciation for the learning opportunities provided by those authors and institutions. At last but not least, our heartfelt thanks to all our family members for caring us and for the huge support.

REFERENCES

[1] Symantec, "Symantec Internet Security Threat Report: Volume XII," Symantec, 2008.
[2] Symantec, "Internet Security Threat Report," vol. 6, 2011.
[3] F-Secure,"F-Secure Reports Amount of Malware Grew by 100% during 2007," http://www.f-secure.com/en_EMEA/about-us.html, Aug.2009.
[4] K. Griffin, S. Schneider, X. Hu, and T. Chiueh, "Automatic Generation of String Signatures for Malware Detection," Proc. 12th Int'l Symp. Recent Advances in Intrusion Detection, 2009.
[5] J.O. Kephart and W.C. Arnold, "Automatic Extraction of Computer Virus Signatures," Proc. Fourth Virus Bull. Int'l. Conf., pp. 178-184, 1994.
[6] A.V. Aho and M.J. Corasick, "Efficient String Matching: An Aid to Bibliographic Search," Comm. ACM, vol. 18, pp. 333-340, 1975.