Malware Detecting System using Mobile Agents

Sayana Catharin Paul, *PG Scholar*
Department of Computer Science & Engineering, Amal Jyothi College of Engineerng, Kanjirappally

Fr. Dr. Biju John, *Faculty*
Department of Computer Science & Engineering, Amal Jyothi College of Engineering, Kanjirappally

Abstract:

Malware is a pervasive problem in distributed computer and network systems. Malware detection and prevention is critical for the protection of computing systems. Identification of malware variants provides great benefit in early detection. Today's AV are unable to detect the malware variants. Control flow has been proposed as a characteristic that can be identified across variants, resulting in classification employing flow graph based signatures. Here we proposed administrator tool to find malwares in the neighbour nodes in network using mobile agents . Here we used flow graph based signature to find malware variants using two matching algorithms: exact matching algorithm and approximate matching algorithm. The exact flow graph matching algorithm uses string based signatures. The approximate flow graph matching algorithm is used to find the variants of malwares. To demonstrate the malware detection, we implement a system and evaluate it using synthetic malwares

Keywords: Malware, control flow graph, Disassembler, mobile agent.

I.INTRODUCTION

Malware short for malicious software means a variety of forms of hostile, intrusive, or annoying software or program code. Malware is a pervasive problem in distributed computer and network systems

Detection of malware is important to a secure distributed computing environment. The predominant technique used in commercial antimalware systems to detect an instance of malware through the use of malware signatures. Malware signatures attempt to capture invariant characteristics or patterns in the malware that uniquely identifies it.

The patterns used to construct a signature have traditionally derived from strings of the malware's machine code and raw file contents [1]. String based signatures have remained popular in commercial systems due to their high efficiency, but can be ineffective in detecting malware variants. Malware variants [2]often have distinct byte level representations while in principal belong to the same family of malware. The byte level content is different because small changes to the malware source code can result in significantly different compiled object code.

Mobile agent is a program that has the ability to move from one node another in network. Mobile Agent is a prominent technology in distributed computing. So it has many applications in system and network administration, information retrieval and e-commerce

In this paper we proposed a malware detecting tool using mobile agents. We proposed control flow graph based signature to find malware variants.

II.BACKGROUND

a). mobile agent

A mobile agent is a software agent that has the additional property that it is not bound to operate only in the system in which it

started. A mobile agent[7] has the unique property that during its lifetime it can be halted, its state and code moved to another computer on the same network, and then continue executing from where it stopped executing on the previous computer. A mobile agent is autonomous because it may decide itself where it will go, what it will do there, and how long it will exist for. However, its environment or other mobile agents may also influence it. Although mobile agents do not provide a solution to any previously unsolvable problems ,they do have advantages over other technologies . They can be used to benefit or to simplify different types of application areas. Some examples of these application areas include ecommerce, distributed information retrieval, telecommunication networks services, and monitoring and notification .

b). Malwares

Malware also known as malicious (or malevolent) software, is software used or created by attackers to disrupt computer operation, gather sensitive information, or gain access to private computer systems. It can appear in the form of code, scripts, active content, and other software. 'Malware' is a general term used to refer to a variety of forms of hostile or intrusive software.

Malware includes computer viruses, worms, Trojan horses, spyware, adware, rootkits Backdoors and other malicious programs. Malware's most common pathway from criminals to users is through the Internet: primarily by e-mail and the World Wide Web.

Figure 1:Types of Malwares

III.EXISTING SYSTEM

Static analysis incorporating n-grams [3,4], edit d instances , API call sequences and control flow [5] have been proposed to detect malware and their polymorphic variants. However, they are either ineffective or inefficient in classifying packed and polymorphic malware.

A malware's control flow information provides a characteristic that is identifiable across strains of malware variants. Approximate matching's of flow graph based characteristics can be used in order to identify a greater number of malware variants. Detection of variants is possible even when more significant changes to the malware

IV.PROBLEM DEFINITION

A malware classification system is assumed to have advance access to a set of known malware. This is for construction of an initial malware database. The database is constructed by identifying invariant characteristics in each malware and generating an associated signature to be stored in the database. After database initialization, normal use of the system commences.

The system has as input a previously unknown binary that is to be classified as being malicious or non malicious. The input binary and the initial malware binaries may have additionally undergone a code packing transformation to hinder static analysis. The classifier calculates similarities between the input binary and each malware in the database. The similarity is measured as a real number between 0 and 1 – 0 indicating not at all similar and 1 indicating an identical or very similar match. This similarity is a based on the similarity between malware characteristics in the database.

If the similarity exceeds a given threshold for any malware in the database, then the input binary is deemed variant of that malware, and therefore malicious. If identified as a variant, the database may be updated to incorporate the potentially new set of generated signatures associated with that variant.

V.PROPOSED SYSTEM

In this paper, we propose administrative tool to find malwares in neighbour nodes in network. Administrator can search the systems to find malwares in LAN from his own station. Mobile Agent technology is used to search neighbour nodes.Mobile Agent is a program code .It has the ability to move from one node to another and perform any task on behalf of it user
Mobile agent has many distributed applications.
In our work, first we create an agent at host A and send the agent to selected node suppose host B. At host B agent search executable files and classify it as malicious or non malicious. If there is any malware, agent send a message back to host A.

To classify an executable file, the analysis makes use of a malware database. The database contains the sets of flow graph signatures, represented as strings, of known malware[6]. To classify the input binary, a similarity is constructed between the set of the binary's flow graph strings and each set of flow graphs associated with malware in the database.

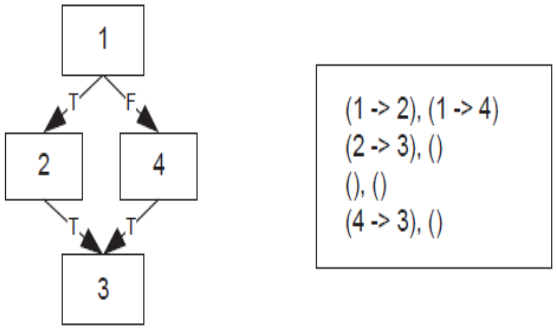

Figure 2: A depth first ordered flow graph and its signature

To classify a Win32 exe file. It has to disassemble using IDA pro disassembler and construct a control flow graph. To generate a signature, the algorithm orders the nodes in the control flow graph using a depth first order, although other orderings are equally sufficient. A signature subsequently consists of a list of graph edges for the ordered nodes, using the node ordering as node labels. This signature can be represented as a string. An example signature is shown in Fig.

To improve the performance, a hash of the string signature can be used instead. MD5 is used here. The advantage of this matching algorithm over approximate matching is that classification using exact matches of signatures can be performed very efficiently using a dictionary lookup.

Malware classification using approximate matches of signatures is performed, and intuitively, using approximate matches of a control flow graph, instead of exact isomorphism's, should enable identification a greater number of malware variants. In our approach we use structuring to generate a signature that enables approximate matching using string edit distances. String pattern matching is used to find the similarity between two strings of malwares. Exact matching can detect the similar signatures

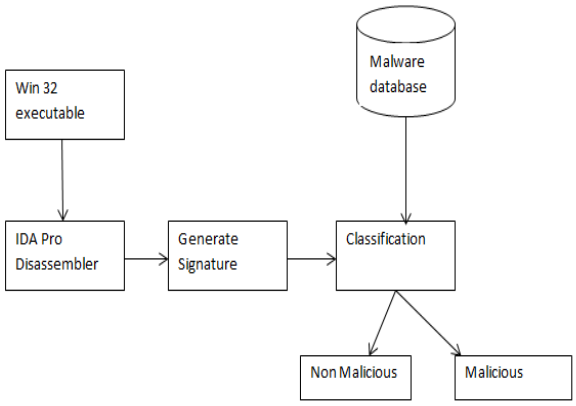

Figure 3: Basic block diagram of malware classification system

VI. CONCLUSION

Malware can be classified according to similarity in its flow graphs. This analysis is made more challenging by Malware variants. We proposed performing malware classification using either the edit distance between structured control flow graphs, or the estimation of isomorphism between control flow graphs. We proposed a system to find malwares in the neighbour node in LAN. We evaluate it using synthetic malwares

REFERENCES

[1]. K. Griffin, S. Schneider, X. Hu, and T. Chiueh, "Automatic Generation of String Signatures for Malware Detection," in *Recent Advances in Intrusion Detection: 12th International Symposium, RAID 2009*, Saint-Malo, France, 2009

[2]. J. O. Kephart and W. C. Arnold, "Automatic extraction of computer virus signatures," in *4th Virus Bulletin International Conference*, 1994, pp. 178-184.

[3].M. Gheorghescu, "An automated virus classification system," in *Virus Bulletin Conference*, 2005, pp. 294-300.

[4]. Y. Ye, D. Wang, T. Li, and D. Ye, "IMDS: intelligent malware detection system," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007

[5]. E. Carrera and G. Erdélyi, "Digital genome mapping–advanced binary malware analysis," in *Virus Bulletin Conference*, 2004, pp. 187-197

[6].T. Dullien and R. Rolles, "Graph-based comparison of Executable Objects (English Version)," in *SSTIC*, 2005.

[7]. Manoj Kumar Kona and Cheng-Zhong Xu, A Framework for Network Management using Mobile Agents.