A Malware Homologous Analysis Method Based on Sequence of System Function

Qiang Guan1, Yong Tang2, Xing Liu3
School of Computer Science, National University of Defense Technology, 410073, Changsha, Hunan, P.R.China
guanqiang21@163.com1, ytang@nudt.edu.cn2, tianwai17@sina.com3

Abstract. The methodology of homology analysis for malware can be used to estimate the phylogeny of malware samples. This paper proposes a malware homology analysis method based on sequence of system function, which is used to solve the problem of universal evolution of malware samples with the help of the bioinformatics tools. The results show that our method can not only be taken as an evolution analysis tool of the same code family, but also be as a malware clustering analysis tools.

Keywords: malware, biology informatics, sequence, phylogeny, clustering

1 Introduction

Modern computer and communication infrastructures are highly susceptible to various types of attack. A common way of launching these attacks is by means of malicious software (malware). As the important foundation of malware detection and prevention, the research on automatic malware analysis has been drawing a lot of concerns. The state of the art malware analysis work usually includes static analysis, dynamic analysis, clustering etc. Several analysis techniques for malware have been proposed.

Static analysis uses program analysis methods of software engineering to analyze the malicious code, such as control flow graph, data flow graph, call graph, the key API map etc[1]. Signature-based methods rely on the identification of unique strings in the binary code which are determined by expert[2] or by machine learning techniques[3]. In dynamic analysis (known as behavioral-based analysis) method is also named as sandbox system, which is based on information collected from the controlled environment (i.e., virtual machine)[4] at runtime, (such as memory usage, the relationship among processes and threads, network behavior)[5]. Clustering techniques identify the same malicious code from the different samples by the hash value of structured information in the executable code, the behavior of malicious code and so on. But when the malicious code's behavior changes with specific conditions (time, system status and environment, etc.), the accuracy of cluster analysis is difficult to guarantee.

1 This work was partially supported by the National Natural Science Foundation of China under Grant NO.61003303
Besides these three methods above, in practical applications, we concern much about the derived evolutionary relationships between malware. Indeed among hundreds of millions of known malicious code, most of them are variations of some old programs[6]. That is to say, the malicious code writers modify, reuse, maintain the codes of malware development libraries and tools package to extend the life cycle of the malware, then this naturally establish a malicious code network as the "evolutionary relationships" in the biology. We can identify, associate, classify and rename malware by the malicious code evolution diagram.

Comparing and adjusting code sequence is an important method of homologous analysis. Recent research including postfix tree, edit distance model and multi-sequence adjust algorithms have two main shortcomings, at the first, enough code sequences should be reserved under the evolution process to form a good evolution model, second for signature-based anti-virus scanners work by searching special code sequence to recognize malware, malware writers can destruct the version sequences to avoid being detected.

This paper proposes a homologous analysis method based on the sequence of system function of malware with bioinformatics tools. Experiment results with 46 malware samples show that our algorithm works well.

As the malware functions are finally implemented by system functions, system function can reflect a code's functions and intrinsic characters. We can represent the malware sample function call diagram by sequence or graph. Since in bioinformatics area there are some relationship evolution models[7] based on nucleic acid, protein or gene sequences, we can use some methods in bioinformatics to generate the malware evolution model[8] to assist the malware analysis, especial for sequence comparing and network comparing. Sequence comparing is to find the common part of multiple sequences, network comparing is also named graph comparing, which compares the similarity between two graphs. In this paper we use the sequence of system function, the algorithms and tools of big protein molecule comparing in bioinformatics area to construct malware evolution diagram to mine the intrinsic nature of malware.

2 A malware homologous method based on sequence of system function

2.1 System framework

In biology, 'evolution relationship' describes the relationship among different species. The automatic analysis tools can infer and reconstruct the evolution model, we use it to construct the malware evolution model. The system framework diagram is illustrated as Fig.1, this paper's main work is marked with a shadow.
2.2 Extraction the sequence of system function

The static character includes control flow diagram, data flow diagram, function call diagram, system function sequence, string sequence, anti-compile code sequence and basic program block etc. We concentrate on construction the function cross-call diagram.

IDA PRO (Interactive Disassembler Professional) is one world class interactive recursive descend anti-compile tool which has complete cross-quotation functions. It generates a database after malware analysis, also provides plugins to access that database programming interfaces which can be used to fetch malware information we needed, such as the function cross-quotation diagram information used in this paper.

Fig. 2. the linear view of address space

![Fig. 2. the linear view of address space](image)

Fig. 3. graph of function xrefs

Fig 2 is the linear view of address space of malware sample generated by IDA, in Fig.2, different colors represents different file types (data, code), we can alert whether the malware has been modified by shelling and other fuzzy methods. In this paper, we only concern about the samples which have been unshelled already. Fig.3 is a graph of function xrefs in a malware sample, we take this diagram as the input of system function sequences extraction algorithm.

Table 1. Extract the sequences of system function

<table>
<thead>
<tr>
<th>input: graph of xrefs from function</th>
<th>output: sequences of system function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Construct a tree structure from the graph of xrefs from function, structure contains</td>
<td></td>
</tr>
<tr>
<td>1.1 function name, string type, set to null if the prefix is 'sub.'</td>
<td></td>
</tr>
<tr>
<td>1.2 nodePtr, pointer type, point to the called</td>
<td></td>
</tr>
</tbody>
</table>
target function child node's linklist

1.3 isLeaf, boolean type, identify whether this node is leaf node or not
1.4 isSys, boolean type, identify whether this leaf node is belong to system functions. (function name with prefix 'sub_' means non-system functions)
1.5 isDFS, boolean type, identify whether all child nodes have been searched

2. Traverse from the root node
if(isLeaf)
begin
backtracking;
if(isSys)
    add name of each child node in the backtrack to its father node's name, set each node's isDFS value
end.
else
begin
if(isDFS)
    backtracking, add name of each child node in the backtrack to its father node's name
else
    traverse nodePtr with depth-first algorithm
end.

3. Sort the name of system function in the root node's name domain, return the sample's system function sequences.

2.3 Normalization of system function sequences and transformation of protein code sequences

Graphic Description Language (GDL) has rich graphic description ability and can clearly define nodes and edges in the graph. Extensible Markup Language (XML) is used to mark electronic document with good structural, it is a language independence of application, easy to exchange and with good and readable structure. We use GDL to describe the function cross-quoting diagram. Fig.4 is a simple gdl example of function cross-quoting diagram, we use XML normalize function call sequences in each non-system function. Fig.5 is an xml example of function cross-quoting sequences. We use GDL and XML to remove redundant and contradictory information, improve information utilization and then reach an integrated and consistent description of malware.
In order to effectively use bioinformatics software tools for homology analysis of malware, the most important issue is unifying the incompatible coding of the two different areas. Since protein sequence is expressed in standard amino acid code, and there has been a FASTA format, which representing the amino acid sequence of text-based file format called FASTA file format which represent amino acid by single letter. In this paper, we transform the sequences of system function to biological protein sequences and transform a letter into two amino acids.

### 2.4 A homologous analysis based on malware function calls

In bioinformatics molecular networks, nodes represent the molecules (genes, proteins, etc.), edges represent the interactions between them. Comparing these networks, there is an important role in disease prevention and treatment, research of the origin and evolution of species etc.

Since complete comparison of two large-scale molecular networks is computationally infeasible, lots of heuristic algorithms for large-scale biological networks are proposed. Similar with the biomolecular networks, malware information (sequence of strings, the sequence of system API call, etc.) can be represented by sequences, some of the information (program basic block, the relationship among process and threads, the implementation of the multi-path behavior, etc.) can be represented by complex high degree of network. We can construct malware family tree by sequences comparing and graph comparing algorithms.

The Unipro UGENE is a mature bioinformatics sequence comparing and large-scale network comparing tools, it is a multi-platform open source software, integrates lots of widely used bioinformatics tools, it can assist people lack of bioinformatics experience to manage, analyze and visualize their bioinformatics data. Fig.6 shows protein sequences representation of a sample set of the sequences of system function.
We can use the evolution toolkit PHYLIP (the PHYLogeny Inference Package) to get the aligned protein sequences, then get the sample's phylogeny tree. PHYLIP uses hidden Markov model (HMM) to infer the evolution of amino acids in the protein.

HMM is based on Markov chain model, which consists of two sequences of random variables, one is unobserved hidden Markov chain, and the other is an observed random sequence, which interact each other by a set of probability. In HMM model, hidden markov chain has three basic states, the match state, insert state, and delete state, also includes start state and end state. Observed sequence value can be A, G, C, T or other value within the 20 kinds of amino acids. A hidden Markov chain can move on the amino acid sequence, comparison by insert or delete operation according to the state.

3 Experiments

3.1 Experiment preparation

We only chose samples which can be unshelled and anti-complied with IDAPro, at last we get 46 malware samples, which have 19 Net-Worm.Win32.Aspxor samples, 14 Net-Worm.Win32.Padobot samples and 13 Net-Worm.Win32.Koobface samples. For convenience, we notate Net-Worm.Win32.Aspxor to xor, Net-Worm.Win32.Koobface to koob and Net-Worm.Win32.Padobot to pbot. All the samples are drown from VX Heavens\(^9\). Our purpose is to effectively distinguish these three sample sets and get the derived evaluation relationships.

3.2 Experiment results

The phylogeny trees of sample sets xor, koob and pbot are illustrated in Fig.7. Region in the two ellipses represent xor and pbot respectively, and the outside is koob malware family.
The closer the distance of malware of the same family in the evolution diagram, the more the similarity do they have, and the less the part that would be modified from one to the other. In Fig.7, xor.ax and xor.ay belongs to the same branch, they locate near each other in the malware family evolution diagram. It fits to the naming rules. Results show that methods based on sequences of system function have a positive effect on the homology analysis of malware.

4 Conclusion and summery

After long terms of evolution, biological characteristics become collections of relatively stable characteristics protein. We can get statistical quantitative and objective species evolution relationships by analyzing large number of samples. Under the existing software structure, the system function is a relatively stable characteristic sequence, this paper takes some algorithms and implementations of bioinformatics homology and evolutionary analysis to analyze malware and construct the phylogeny trees of a set of malicious codes. From the experimental results, it can be seen that the sequence homology analysis methods provide effective technical tools and open source tools have important theoretical and practical value for the depth of understanding and prevention of malicious code and evidences for computer crime.

References