Host Security Based Research on Automatic Generation
Method of Security Threat Intelligence

Zhuang CHEN¹,a and Jun-jie CHANG²,b

¹ College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China,
²College of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

¹cz@cqut.edu.cn, bchang_junjie1991@163.com

Keywords: Security threat intelligence (STI); Malware family clustering; Behavior correlation analysis.

Abstract. In order to conduct in-depth analysis on the operating principle of malware and reduce its analysis period, this article proposes a behavior analysis based malware family clustering method; it conducts deeper layer behavior correlation analysis according to sample selection of different families, to generate the Security Threat Intelligence representing for the malwares of this family. It is beneficial to the rapid expansion of antimalware engine behavior library, and can give active response to host computer security events, to detect malware variants. According to experimental findings, this method can extract effective Security Threat Intelligence, detect malware quite effectively and efficiently.

Introduction

According to “China’s Internet Security Report of 2016” released by 360 Internet Security Center [1], the increasingly frequent network attacks such as APT are leading to stealing of confidential information of governments and enterprises, destroying of industrial systems and economic loss of financial systems. The formation of cooperative security monitoring, response and in-depth defensive system through Security Threat Intelligence (STI) driving is the only way to effectively cope with such advanced threats.

STI includes a wide range of contents, which is still in developing stage, with a variety of explanations. In general, STI is the knowledge based on evidence knowledge including context, mechanism, influence and suggestion for answering, aiming at an existing or emerging threat or behavior harmful to asset, used to help solve threat or harm. Most unknown viruses evolve from known viruses; massive malware variants have different signature but similar behaviors; therefore, most malware signature based antimalware engines are always in hysteretic state. The literature [2] proposes the clustering based malware behavior research to judge malware or safe files, to detect malware. However, this method fails to take the complexity of malware into consideration; due to the lack of expandability, it leads to high misjudgment rate when facing with massive samples. Sami and Sathyanarayan et al. [3][4] proposes to judge malware by utilizing API calling sequence. However, this method fails to conduct in-depth analysis on the aspect of behavior correlation; besides, due to the high layer of API calling, the computer has been captured when the malware API calling sequence is detected.

Based on the monitoring on the behavior of the driver layer, this article abstracts malware behaviors, and then conducts clustering analysis by collecting malware family
behaviors; it then conducts deeper layer behavior correlation analysis by utilizing malware samples which can generalize the family, to finally generate the STI, and verify the validity.

**Malware Monitoring**

Malware or normal program behavior can be defined as an event, for it expresses certain action finished by a program. Totally 34 behavior monitoring items are customized, which are behavior characteristics. Mas' et al. [5] performing feature selection can improve detection efficiency.

According to conclusion and abstraction on massive windows behaviors, the kernel characteristics of events are composed by the following four parts:

1) **Event function description**: the functions can be realized by the event.
2) **Event object information**: the operation object of the event.
3) **Event characteristic data information**: the auxiliary data information realized by the event.
4) **Event time attribute**: used to confirm the sequences of events.

Therefore, event can be defined as follows: Events, as for the specific \( n, o, f, t \) \( \in \) Events, \( n \in \) Events\( \text{Name} \), \( o \in \) ObjectInformation, \( f \in \) FeatureData, \( t \in \) TimeInformation; according to the definition, the event with the name of \( n \) has the parameter object information of \( o \), and the characteristic data of \( f \), and the time sequence information during the code executing process of \( t \).

**Malware Analysis by Clustering**

**Characteristic Extraction**

Firdausi et al. [6] makes comparisons between four classification algorithms of kNN, Naïve Bayes, SVM and J48, to detect malware based on behavior analysis. The data characteristic dimension in this literature is composed by behavior terms, which constitute a dictionary, to accumulate the occurrence times of each behavior term during malware monitoring, which is the characteristic value on the dimension. The data has the characteristic of high latitude, not applicable to DBSCAN algorithm. According to research [7] findings, DBSCAN is an suitable algorithm; therefore, it is needed to construct the matching characteristic set, the Jaccard Distance is adopted here for the measurement of abstract type values.

Jaccard similarity coefficient is used for making comparisons on similarities between abstract data points, which is the most widely utilized similarity algorithm. This algorithm is defined as the proportion of the number of the elements in two combined intersections in the union set. In condition that there is no repeated element in the two sets, the coefficient is 0. If the two sets have totally the same elements, the coefficient is 1. The calculating formula of coefficient is as follows:

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}.
\] (1)

Jaccard Distance measures the distinction degree of two sets with the proportions of different elements in all the elements in the two sets, which is the reverse concept to Jaccard similarity coefficient, with the following formula:
\[ J_\delta = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \] (2)

**Division on Malware Families**

The algorithm DBSCAN has the following main idea: as for any point \( p \) in a type (boundary points are not included), the number of data points in a given Eps domain is no less than \( \text{Minpts} \). The Eps domain of point \( p \) is defined as \( N_{Eps}(p) = \{q\} \). In which \( \text{dist}(p, q) \) is the distance between point \( p \) and point \( q \); \( \text{Minpts} \) is the minimum number of the points must be included in Eps domain of a point of the class.

Now it is needed to input \( \text{Minpts} \) and Eps. The larger the value of the radius of Eps, the less limitation on the formation of clusters, and the examples with moderate distance may be divided into the same cluster. When \( \text{Minpts} \) is set as 1, it means that the noise point changes to a cluster, which is beneficial to the division of malware family with small number of samples.

This article defines purity, to depict the accuracy of the method, with the following definition:

\[ \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{\phi} \max \frac{A_i \cap B_j}{A_i \cup B_j}. \] (3)

In which set \( A \) is the test output; set \( B \) is the standard; \( m \) represents for the number of clusters of the test output; \( n \) is the number of the class of the original family.

This article utilizes 353 known samples as the test input, which can be divided into 5 classes of malware families. Through parameter adjustment, Eps is set as 0.2, and \( \text{Minpts} \) is set as 1, with the following classification results is shown in Fig.1:

As shown in Fig.1, the samples are divided into 6 classes according to DBSCAN algorithm.

Table I shows the purity of the 6 malware families in the test results.
Table I. Purity Table of Test.

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num</td>
<td>43</td>
<td>167</td>
<td>29</td>
<td>53</td>
<td>58</td>
<td>3</td>
</tr>
<tr>
<td>P</td>
<td>0.94</td>
<td>0.95</td>
<td>0.86</td>
<td>0.88</td>
<td>0.06</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>0.06</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>1</td>
<td>0.03</td>
</tr>
</tbody>
</table>

In Table I, the first line shows the cluster ID; the second line shows the size of each cluster; each of the following lines shows the proportion of the malware of each family in the composition of the cluster. For example, the cluster with ID of 2, has the purity of 0.92 and the size of 167, illustrating that this cluster is composed by 157 malwares of family A and 10 malwares of family B. Cluster 6 belongs to no family.

Therefore, calculate the average purity after getting rid of the purity of cluster 6, with the value of 0.93. It verifies that this clustering method can achieve a high accuracy.

Extraction of Typical Samples

According to the handling of samples in the previous section, the sample set with classification marking information. The main reason for classifying malwares is to select the typical samples of the class for further analysis, to improve the matching rate aiming at the malwares of the family.

The typical sample extraction formula is as follows:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \max \left( \frac{A_i \cap B_j}{A_i} \right).$$

Equation (4) represents the smallest sample of the Jaccard Distance in a cluster. There are high similarity between the selected samples and other samples, with representative characteristics.

Behavior Characteristic Correlation Analysis

Behavior Correlation Tree Model and Algorithm Design

The purpose of behavior correlation analysis is to improve the description ability to malware STI, and reduce the mistake rate. Zhang et al. [8] introduce association rules is efficiently to detect malware.

In consideration that different events include different object information; according to input and output directions, it can be divided into quotation object information and creation object information. According to definitions of events, Object Information can be defined as object information set; as for the object information set of an event

$$ObjInfo = ObjInfo_{in} \cup ObjInfo_{out}$$

$ObjInfo_{in}$ represents for the set of the quotation objects of the event, and $ObjInfo_{out}$ is the set of the output objects of the event. The object information of the event can be divided into the following 5 kinds of conditions is shown in Table II.
Table II. Distance Matrix.

<table>
<thead>
<tr>
<th>Event type</th>
<th>ObjInfo in</th>
<th>ObjInfo out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quotation type</td>
<td>Nonvoid set, with quotation object</td>
<td>Void set, without output object</td>
</tr>
<tr>
<td>Quotation and creation type</td>
<td>Nonvoid set, with quotation object</td>
<td>Nonvoid set, with output object</td>
</tr>
<tr>
<td>Creation type</td>
<td>Void set, without quotation object</td>
<td>Nonvoid set, with output object</td>
</tr>
<tr>
<td>Ending type</td>
<td>Nonvoid set, with quotation object</td>
<td>Void set, without output object</td>
</tr>
<tr>
<td>Independent type</td>
<td>Void set, without quotation object</td>
<td>Void set, without output object</td>
</tr>
</tbody>
</table>

The creation type event creates certain object information, which is quoted by follow-up event. The quotation and creation type event is a transitional creation event, which creates new logic object information while quoting certain object information. The quotation type event is the most common event, which operates with object information, without creating new object information. The ending type event expresses the ending of the present object operation. The independent type event does not involve in object information, which can express certain information independently.

The article constructs the correlation structure of events by adopting the correlation information tree. Correlation information tree is correlation data structure designed in combination of program implementation process characteristics; the root node of the correlation tree represents for current executable program. As shown in the Fig.2.

The algorithm procedures are as follows:

Step 1: Read the event data, and extract the event information; if the data processing is finished, turn to step 4; or turn to step 2.

Step 2: Execute corresponding operation according to event type; in condition of quotation type, search for object information node and judge if there is finishing marking; if it can be searched, and add it as the child node; if it cannot be searched, or there is ending marking, add the object node under the root node, and add the event under the node as the child node.

Step 3: Turn to step 1 to continue to read the data.

Step 4: Translate the correlation tree model into the STI of XML format.
Sample Matching Test

In order to prove the reasonability of this scheme, the two malware family samples and a normal program set sample are utilized for matching test. Test the sample matching rates before and after the behavior correlation, respectively. The behaviors before the correlation are independent; the sample with one matched event is regarded as malicious. The behaviors after correlation are dependent, and the matching can be achieved after finishing node searching and achieving leaf node.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Family A</th>
<th>Family B</th>
<th>Normal sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of samples</td>
<td>167</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>Matching rate before</td>
<td>93%</td>
<td>100%</td>
<td>26%</td>
</tr>
<tr>
<td>correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matching rate after</td>
<td>71%</td>
<td>83%</td>
<td>5%</td>
</tr>
<tr>
<td>correlation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As shown in the Table III, after the combination with behavior correlation analysis, the STI generation method designed in this article can reduce the mistake rate to 5% from 26%; however, it also leads to the matching rate.

Conclusion

This article introduces a STI automatic generation method aiming at massive malware. Firstly, by utilizing the driving based behavior monitoring technology, it acquires malware behavior log, and defines each behavior as an event. Secondly, it conducts research on the similarity between samples; the JaccardDistance is taken as the input of clustering modules; it then makes comparisons between clustering algorithms, and selects DBSCAN. Finally, it designs the behavior correlation tree model, and extracts final STI. According to experimental findings, this method has low mistake, but with unfavorable matching rate.

In consideration that STI includes the information of large number of individual samples, with multiple characteristic dimensions, the work in later period will be conducted mainly on the selection of characteristics, to improve the matching rate of malwares under the premise of ensuring low mistake.

References


