International Journal of Security and Its Applications
Vol. 10, No. 5 (2016) pp.1-16
http://dx.doi.org/10.14257/ijsia.2016.10.5.01

A Novel Information Fusion Model for Assessment of Malware Threat

Chao Dai, Jianmin Pang, Xiaochuan Zhang, Guanghui Liang and Hong Bai

State Key Lab of Mathematical Engineering and Advanced Computing of China
daichaochn@aliyun.com

Abstract

It is not only important for security analysts to judge some binary code is malicious or not, but also to understand the malware "what to do" and "what's the impact it posed on our information system". In this paper, we proposed a novel information fusion model to quantitate the threat of malware. The model consists of three levels: the decision making level information fusion, the attribute level information fusion and the behavior level information fusion. These three levels portray special characteristics of malware threat distributed in the assessment model. Combined with the static analysis technology and real-time monitor technology, we implemented a framework of malware threat assessment. The experiment demonstrates that our information fusion model for malware threat assessment is effective to quantitate the threat of malware in accuracy and differentiation degree. In the end, we discussed several issues that could improve the performance of the model.

***Keyword**: information fusion, malware analysis, threat assessment, static analysis, real-time monitor*

1. Introduction

Information technology plays more and more important role in various areas. Meanwhile the challenge faced by information security increased day by day. The competition on information access, exploitation and control is white-hot. Therefore, it is important to study information security. For a long time, the malware has been one of the major threats to information security. [1] The threat of malware reflected in the 3 aspects as follows:

In scope, the threat of malware is ubiquitous. From personal computers to enterprise servers, all information devices became targets of malware.

In the degree of impact, the consequences of malware are serious. Malware could affect both national strategies and common person's daily life.

In the time span, the evolution of malware never pauses. Although anti-virus systems developed at the same time, the quantity of malware is increasing and the developers of malware are keeping digging tricks to evade the detection of anti-virus systems.

Assessment of malware threat is to reflect comprehensively and accurately what impact will malware pose on users and information systems in the practical computing environment. It is an important issue in the field of malware analysis. Assessment of malware threat has significant application in the information system security situational awareness, information system attack warning and information system attack response. It could optimize the utility of human resources and financial resources to improve the efficiency and the quality of information security. Out of its important role in network security, lots of network security vendors pay attention to the assessment of malware threat.[2][3][4]

At present, the malware threat assessment has the following technical problems need to be solved:

ISSN: 2005-4262 IJGDC
Copyright ⓒ 2016 SERSC

The data source of assessment is relatively simplistic. Therefore, it is difficult to describe the malware comprehensively. Traditional malware threat assessment is mainly based on the disassemble instruction sequences or function call sequences. If malware behavior manifests in different forms, then the single source of assessment data are difficult to characterize the behavior of malware, which cause the incompleteness of the data in assessment;

The assessment algorithm is relatively simple. So it is difficult to quantitate the threat of malware accurately. The threat of malware has its own special characteristics. For example, there are several different implementations for a certain malware's key attribute. However, the influences of these implementations on malware's performance are not cumulative. The key attribute of malware is mainly affected by the optimal implementation. But the traditional assessment algorithm is hard to portray this point;

The assessment model is relatively ambiguous. It is difficult to assess the relationship between assessment elements in fine grain. The traditional assessment models are often evaluated with a single model, such as the architecture evaluation model, the attack tree model *etc*... Whereas the characteristics of malware threat assessment lies on the decision making level, attributes level and behavior level are different, which may not suitable for use single model.

Therefore, we need to research the model of malware threat assessment to solve the above problems, which could provide powerful technical support for the information system security situational awareness, information system attack warning and information system attack response *etc*.. In this paper, we proposed a novel information fusion model for assessment of malware threat which takes full consideration of characteristics of malware threat assessment. To achieve the goal, we implemented a hybrid framework of malware analysis which could obtain comprehensive view of the malware. To analyze the malware in-depth, we draw support from static analysis, dynamic analysis and other methods.

The rest of the paper is organized as follows. Section 2 introduces the overview of the information fusion model. In Section 3, each level of the information fusion model is discussed. Section 4 introduces the architecture of malware detection framework. Section 5 describes the deployment of experiment environment and the result of experiment. Section 6 provides a brief overview of the previous work related to the assessment of malware threat. Section 7 discussed the future work. Finally, section 8 concludes our work.

2. Overview of Model

Figure 1 shows the overview of the information fusion model. The basis of information fusion model labeled as layer L0 is the output of code analysis, which is obtained through static analysis and real-time monitor. Its contents include the file structure information, the disassemble instruction sequence, the function call sequence and the IRP(I/O Request Package) sequence. The details of obtaining the result of code analysis will be discussed in Section 4.

The contents from layer L1 to layer L3 constitute the main body of the information fusion model. Based on their different functions, we call them behavior level information fusion, attribute level information fusion and decision making level information fusion.

Layer L1 fuse the disassemble instruction sequence, the function call sequence, the IRP sequence and other original data of analysis together. Then it summarizes the behavior of code from diverse angles, which complementary with each other to acquire the real behavior of malware.

On the basis of behavior, layer L2 cluster the operations on objects in host, such as process/thread, file, registry, service, drive, network, system resources *etc*... Depend on the operation against different object, the operation was come under a certain attribute that reflect the key capacity of malware. According to the different types of malware, attribute also needs to be adjusted accordingly. For example, with regard to trojan horse, its key attributes includes initiation, concealment, self-protection, resource consumption and others. Whereas for worm, the key attributes consists of initiation, concealment, resource consumption, propagation *etc*..

Ultimately, the attributes obtained in layer L2 are synthesized in layer L3. Then the model could score the threat of a malware.

Based on the score in layer L3, Layer L4 could grade the threat in a clear fashion.

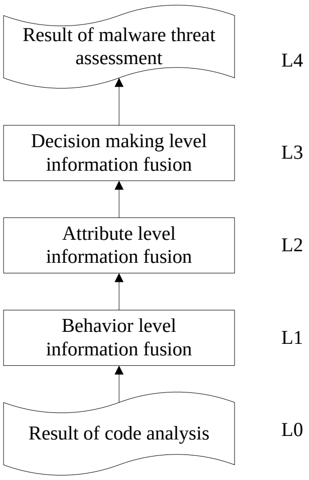
Result of malware threat assessment
L4
Decision making level information fusion
L3
Attribute level information fusion
L2
Behavior level information fusion
L1
Result of code analysis
L0

Figure1. Overview of the Information Fusion Model

In this paper, we take the trojan horse as an example to illustrate our model. The threat of worm and virus could be assessed in a similar way. Figure 2 shows the information fusion model of trojan horse's threat assessment.

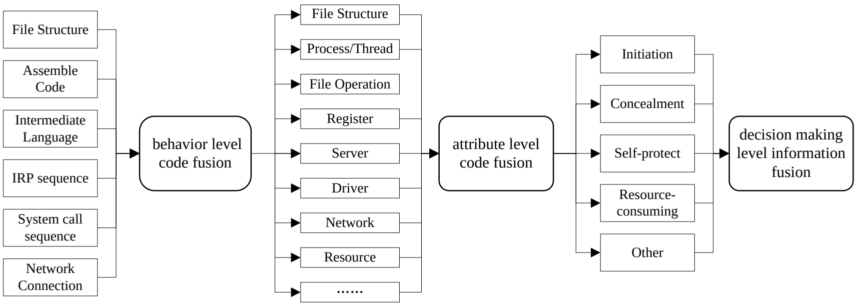

Figure 2. The Information Fusion Model of Trojan Hose

3. The Design of Information Fusion Model

As mentioned above, the core of the model consists of three parts. In order to keep the assessment comprehensively, accurately and thoroughly, each part's algorithm is designed directed against its characteristics.

3.1 Decision Making Level Information Fusion

Malware malicious of decision making level information fusion is to score the threat based on the assessment of the key attributes of information fusion on malware attribute level comprehensively. Decision making level information fusion is the ultimate objective of malware threat assessment.

Copyright ⓒ 2016 SERSC

Decision making level information fusion process has to consider the following important issues:

- The threat of malware is determined by its key attributes;
- The significance of the different attributes differs, so the influence on threat differs.

Decision making level information fusion can be represented by the formula below, where S_{threat} represents the calculated ultimate threat of malware, $S_{Attribute_i}$ represents the i_{th} key attribute of a certain type of malware, fun_D represents the decision making level information fusion function:

$$S_{threat} = fun_D\,(\,S_{Attribute_1}, \ldots, S_{Attribute_n}\,) \tag{3.1}$$

In order to reflect significance of different attributes, we use analytic hierarchy process (AHP) [2]to calculate the relative importance of each attribute to get the weight of each attribute, which noted as $\omega=(w_1,w_2,\cdots,w_n)$. The weights reflect the relevant attributes influence the threat to what extent. Finally, the weight vector with the key attribute fusion value is multiplied to obtain assessment threat, *i.e.*:

$$S_{threat} = \left(w_1, w_2, \ldots, w_n\right)\cdot\left(S_{Attribute_1}, \ldots, S_{Attribute_n}\right)^T \tag{3.2}$$

Next, we take the trojan horse threat assessment as an example, the key attributes including initiation, concealment, self-protection, resource consumption and other. The steps using the AHP to determine the weights as follows:

(1) Construct the judgments matrix with the results of pairwise comparison of key attributes.

$$M = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \left(a_{ij} > 0, a_{ji} = \frac{1}{a_{ij}}, a_{ii} = 1\right) \tag{3.3}$$

The judgment matrix M_{trojan} of trojan horse with five attributes is:

$$M_{trojan} = \begin{bmatrix} 1 & 1/7 & 1/5 & 3 & 3 \\ 7 & 1 & 5 & 7 & 7 \\ 5 & 1/5 & 1 & 3 & 3 \\ 1/3 & 1/7 & 1/3 & 1 & 1 \\ 1/3 & 1/7 & 1/3 & 1 & 1 \end{bmatrix} \tag{3.4}$$

(2) The sorting of significance. According to the judgment matrix, calculate the eigenvectors ω corresponding to the largest eigenvalue of λ_{max}. Equation as follows:

$$M\cdot\omega = \lambda_{max}\cdot\omega \tag{3.5}$$

Then the eigenvector ω was normalized. The normalized eigenvector ω is corresponding to the weight of each attribute.

$$\omega=(w_1,w_2,\cdots,w_5\,) \tag{3.6}$$

According to the judgment matrix M_{trojan}, it has the property:

$$M_{trojan}\cdot\omega = \lambda_{max}\cdot\omega \tag{3.7}$$

Then we got

λ_{max} =5.4378, and ω=(0.1019,0.5745,0.2145,0.0546,0.0546)

(3) The consistency check. Whether the weight distribution above is reasonable, it also needs to check the consistency of judgment matrix. Test using the formula:

$$\mathrm{C.R.}=(\mathrm{C.I.})/(\mathrm{R.I.}) \tag{3.8}$$

Among them, C.R. is a random consistency ratio of judgment matrix; C.I. is a general consistency of judgment matrix. It is given by:

C.I.=(λ_{max}-n)/(n-1) (3.9)

R.I. indicates means random consistency index of judgment matrix. The R.I. values of 1~11 order judgment matrix shown in table 1.

Table 1. 1 to 11 Order Matrix with Random Consistency Index Table

n	1	2	3	4	5	6	7	8	9	10	11
R.I.	0	0	0.52	0.90	1.12	1.26	1.36	1.41	1.46	1.49	1.52

For the trojan horse threat assessment matrix M_{trojan}, we have λ_{max}=5.4378. So we calculated C.I.=(5.4378-5)/(5-1)=0.1095<1.12, which satisfies the consistency. Therefore, it is reasonable to acquire the weight of key capability through AHP.

Finally, the vector of weight of key attribute multiply with attribute level information fusion is the results of the assessment of the threat of the Trojan, *i.e.*:

$$\left(w_1,w_2,w_3,w_4,w_5\right)\cdot\left(S_{initiation},.S_{concealment},S_{protect},S_{resource},S_{other}\right)^T \tag{3.10}$$

$$S_{trojan\_threat}=\left(0.1019,0.5745,0.2145,0.0546,0.0546\right)\cdot\left(S_{initiation},.S_{concealment},S_{protect},S_{resource},S_{other}\right)^T \tag{3.11}$$

3.2 Attribute Level Information Fusion

The attribute level of malicious information fusion is to quantitate the impact to a certain attribute of relevant acts based on malware behavior analysis. Moreover, according to the different types of malware, attribute needs to be adjusted as well. Attribute level information fusion is the core of malware threats assessment.

Attribute level information fusion process needs to consider the following key issues:

- A certain attribute is determined by its implementations;
- The more implementations, the stronger of capability of relevant attribute. Take the attribute of initiation as an example, the implementation consist of modifying the configuration file, modifying the registry, loading system service *etc*... As the number of implementation growing, the intent of relevant attribute could be realized with higher probability;
- If a certain attribute has more than one implementation, then the attribute is determined by its optimal implementation. Take the attribute of initiation as an example. Compared with modifying the configuration file, modifying the registry implementations is easy to be detected. But if implementations include loading system service, then we think the initiation attribute is mainly decided by the loading system service.

Attribute level information fusion can be represented by the formula, wherein $S_{Attribute_i}$ represents the i_{th} key attributes of a certain malware. $Type_{ij}$ represents the j_{th} implementation of the i_{th} key attributes. fun_A represents the attribute level information fusion function:

$$S_{Attribute_i}=fun_A\left(Type_{i1},...,Type_{in}\right) \tag{3.12}$$

According to the key issues mentioned above, we adopt norm to measure the threat of a specific attributes. w_{ij} represents the weight of the j_{th} implementations for the i_{th} key attributes, which obtained by expert experience. The sum of the weights of implementation of i_{th} key attributes equal to 1. If the attribute using the j_{th} implementation, then $Type_{ij}$ =1, otherwise $Type_{ij}$ =0. That is:

$$S_{Attribute_i} = \sqrt{\sum_{j=1}^{n}\left(w_{ij} \cdot Type_{ij}\right)^2} \text{ , } 0 \leqslant w_{ij} \leqslant 1\text{, } \sum_{j=1}^{n} w_{ij} = 1\text{, } Type_{ij} \in \{0,1\} \tag{3.13}$$

Take the trojan threat assessment as an example, $S_{initiation}$ represents the fashion of trojan horse to initialization:

$$S_{initiation} = \sqrt{\sum_{j=1}^{n}\left(w_{ij} \cdot Type_{ij}\right)^2} \text{ , } (0 \leqslant w_{ij} \leqslant 1\text{, } \sum_{j=1}^{n} w_{ij} = 1\text{, } Type_{ij} \in \{0,1\}) \tag{3.14}$$

$S_{concealment}$ represents the concealment attribute of trojan horse:

$$S_{concealment} = \sqrt{\sum_{j=1}^{n}\left(w_{cj} \cdot Type_{cj}\right)^2} \text{ , } (0 \leqslant w_{cj} \leqslant 1\text{, } \sum_{j=1}^{n} w_{cj} = 1\text{, } Type_{cj} \in \{0,1\}) \tag{3.15}$$

$S_{protect}$ represents the self-protect attribute of trojan horse:

$$S_{protect} = \sqrt{\sum_{j=1}^{n}\left(w_{pj} \cdot Type_{pj}\right)^2} \text{ , } \quad (0 \leqslant w_{pj} \leqslant 1\text{, } \sum_{j=1}^{n} w_{pj} = 1\text{, } Type_{pj} \in \{0,1\}) \tag{3.16}$$

$S_{resource}$ represents the resource-consuming attribute of Trojan horse:

$$S_{resource} = \sqrt{\sum_{j=1}^{n}\left(w_{rj} \cdot Type_{rj}\right)^2} \text{ , } \quad (0 \leqslant w_{rj} \leqslant 1\text{, } \sum_{j=1}^{n} w_{rj} = 1\text{, } Type_{rj} \in \{0,1\}) \tag{3.17}$$

And S_{other} represents the other attribute of Trojan horse:

$$S_{other} = \sqrt{\sum_{j=1}^{n}\left(w_{oj} \cdot Type_{oj}\right)^2} \text{ , } (0 \leqslant w_{oj} \leqslant 1\text{, } \sum_{j=1}^{n} w_{oj} = 1\text{, } Type_{oj} \in \{0,1\}) \tag{3.18}$$

3.3 Behavior Level Information Fusion

Malware behavior level information fusion takes advantage of static and real-time monitor methods of analysis of malware to find the operation on objects including file structure, process/thread, file, registry, service, drive, network, system resources. Then we could obtain the operation of the objects relates to the object file structure, assembly instructions, function calls, IRP sequence and other information, to judge the behavior of malware. The behavior level information fusion is the basis of malware threat assessment.

Although the arrangement of disassembly instructions sequence, the function call sequence and the IRP sequence is varied, we extract the relevant sequence corresponding to a certain object mentioned in 3.2 as the basis for further analysis.

The behavior level information fusion process needs to consider the following key points:

- The code characteristic, implementation method and the behavior of code are parallelism. (That is to say a certain implementation method have corresponding code characteristics, and behaviors);
- Attributes of a certain implementations can be achieved in a variety of ways. (e.g. the initiation attribute of trojan horse);
- Different code characteristics require distinct behavior decision algorithm.

According to the key issues mentioned above, $Type_i$ represents a certain implementation, $Behavior_{in}$ indicates the n_{th} implementation of i_{th} behavior. fun_B represents the behavior level information fusion function. A certain implementation is found in the code, if it satisfies the formula below:

$$Type_i = fun_B\,(Behavior_{i1},\ldots,Behavior_{in}) \tag{3.19}$$

or

$$Type_i = Behavior_{i1} \& \cdots \& Behavior_{in} \tag{3.20}$$

For a certain behavior $Behavior_{ij}$, with I_ASM indicating the disassemble instruction sequence, I_FUN representing the function call sequence and I_IRP indicating the IRP sequence. Due to it is inevitable that code behavior embodied in either disassemble instruction sequence, or the function call sequence and IRP sequence. So once we find the corresponding characteristic in a certain aspect, we could conclude the code take on the corresponding behavior, *i.e*.

$$Behavior_{ij} = \text{I\_ASM} \,|\, \text{I\_FUN} \,|\, \text{I\_IRP} \tag{3.21}$$

4. A Hybrid Framework of Malware Analysis

In order to characterize malware accurately and comprehensively, we designed a hybrid framework called *Libra*, which consists of static analysis module and real-time monitor module. Figure 3 shows the architecture of malware analysis hybrid framework. The static analysis module analyzes the code file structure, the disassembly and the control flow graph to obtain the result of static analysis. The real-time monitor module consists of IRP monitor, kernel monitor and network monitor. The result of static analysis and the result of real-time monitor information are fused together to determine the existence of a behavior.

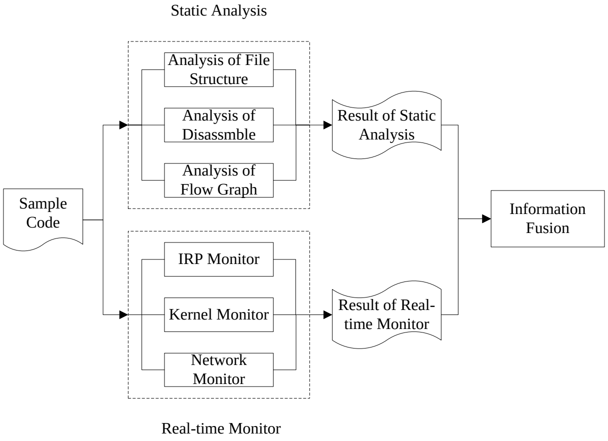

Figure 3. The Architecture of Malware Analysis Hybrid Framework

4.1. Static Analysis Module

Static analysis module consists of three tasks: analysis of file structure, analysis of disassembles and analysis of flow graph.

Analysis of file structure: malware always has some wired features in the file structure, which could be used as heuristic rules. These rules include: the suspicious field of file header, the size of PE optional header and the gap between adjacent section *etc*..

Analysis of disassemble: the instruction sequence could reflect program original intention which always been the important indication of code behavior. [6] However, out of malware always apply anti-anti-virus tricks to defeat the disassemble. So there are many algorithms to decrease the effect of anti-anti-virus tricks. In addition, the battle between malware writer and malware analyzer will be more white-hot in the future. [7][8]

Analysis of flow graph: out of most malware writer apply anti-anti-virus tricks to protect themselves, which causes the analysis of disassemble is prone to erroneous. Nevertheless, the structure of control flow and data flow is relative stable, so the analysis of control flow graph and data flow graph could help the analysis of malware. [9][10]

4.2. Real-time monitor module

Real-time monitor module consists of three tasks: IRP monitor, Kernel monitor and Network monitor.

IRP monitor: IRP packet is an important data structure in device driver. The applications perform file operation need to send IRP packets to drivers. Accordingly, IRP monitor could keep watch on the file operation, which is the target of plenty of malware.[11][12]

Kernel monitor: malware work has to call kernel function to do malicious operations. So it is necessary to keep track of system function call. It is strong certification of malware behavior.[13][14]

Network monitor: Besides the host behavior, nowadays some malware, such as worm and trojan horse, always manifest in the network.[15] Take trojan horse for example, in order to keep attract with the controller, it has to send packets to controller or some website to prove it is still working. In addition, the worm in the process of propagation, the massive packets it sends over the network is different from the normal packets.

5. Experiments and Results

5.1. Deployment of Experiment Environment

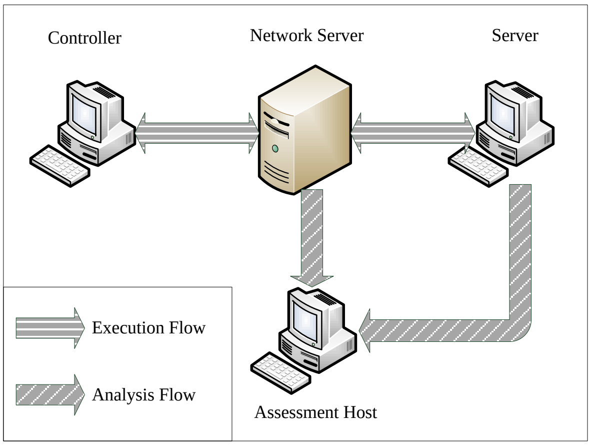

Figure 4. The Deployment of Experiment

Figure 4 shows the deployment of experiment environment. In the model of deployment, the experiment environment comprises the controller host, the server host, Network Server and Assessment host. The controller host is mainly responsible for the installation of trojan controller and run the trojan on the controller host. The server host deploy host behavior acquisition module to gather host relevant data. The network server deploys the network behavior acquisition module to collect the network relevant data. The assessment host is responsible for gathering the host behavior data and the network behavior data to perform the threat assessment.

5.2 Experiment Analysis

We choose three indexes to evaluate the performance of the model. The samples in the experiment are collected from online. In this paper we focus on the threat assessment of trojan horse. With regard to the virus and worm, the assessment could be carried on in the similar way.

(1) Accuracy

(1). Accuracy

First, it is our principal objective for our model to excavate the real behavior of malware. Only understanding what the malware was doing, can the information model get the accurate result.

We use Fireeye [16] as the third-party tool to prove the accuracy of Libra. Fireeye is an online automated malware dynamic analyasis system maintain by Cheetah Mobile Inc., which is similar to the online sandbox of Comodo. It could analyze the behavior of malware in fine grain.

$N_{i_{Third}}$ indicates the number of behavior identified by the third party.

$N_{i_{Libra}}$ indicates the number of behavior identified by our system detection.

$N_{i_{Max}}$ indicates the max number of $N_{i_{Third}}$ and $N_{i_{Libra}}$.

We use P to indication the performance of detection, and it is calculated by the formula below:

$$P={N_{i_{Libra}}}\Big/{N_{i_{Max}}} \tag{5.1}$$

We assume that if $N_{i_{Third}} = N_{i_{Libra}}$ =0, then P=1.

Table 2. The Relative Detection Rate of Libra and Fireeye

Attribute		Libra #	Third-party	P
Initiation	Config-file modifying	0	0	100%
	Register modifying	9	6	100%
	Server loading	22	14	100%
Concealment	Process/Thread concealment	7	13	54%
	File concealment	26	25	100%
	Port concealment	7	0	100%
	Protocol concealment	0	0	100%
	Traffic concealment	32	0	100%
Self-protection	Process ward	0	0	100%
	File bacup	23	16	100%
	Packer	8	0	100%
	Encryption	14	0	100%
Resource	CPU consuming	32	0	100%
	Memory consuming	32	0	100%
	Bandwidth consuming	32	0	100%
Other	Heartbeat	29	0	100%
	rebound	2	0	100%
	Self-unload	16	0	100%
	File cleanup	19	0	100%

Table2 reveals that compared with the third-party analysis system, *Libra* could get the correct result, which provide a solid basis for further assessment. But according to the result, *Libra* need to be improved in the detection of process/thread concealment.

(2) Differentiation degree

The differentiation degree in experiment decides the experiment’s effectiveness, which describe the ability to distinguish of the model.

Table 3. The Score of Trojan Horses

No.	Name	Overall	Initiation	Concealment	Self-protection	Resource consuming	Other
1	Bandook	25	75	20	0	73	30
2	Byshell	37	80	40	0	100	15
3	ConsoleDevil 1.2 serAR	61	80	51	85	80	25
4	FinalFantasy_ DI	72	80	62	98	73	55
5	Freerat2.0	49	80	28	82	46	90
6	Freerat2.0_D	62	80	48	82	73	90
7	gh0st	72	100	61	98	40	62
8	PcShare	40	80	28	50	40	50
9	firefly	64	80	51	92	66	55
10	TB Dark fairy	50	50	40	70	80	55
11	x-door-A	62	80	48	92	100	25
12	x-door-B	62	80	48	92	100	25
13	Hot angle 4.0Server	68	100	58	70	100	70
14	Cold moon	74	90	64	99	93	25
15	Green light	52	50	40	85	73	25
16	Commander_DR	69	40	66	99	73	40
17	Commander _DRS	71	80	62	99	73	40
18	Magic_server	59	80	40	98	100	30
19	Magic_server_P	59	80	40	98	100	30
20	Renwoxing_F	63	80	45	89	93	80
21	Renwoxing_FPA	56	80	33	99	93	55
22	Calf 1.1server	48	96	40	50	66	25
23	crossbow	51	100	33	75	73	25
24	NoResume	60	70	58	70	73	15
25	nuclear	42	75	33	40	80	40
26	ruser	49	80	50	40	66	0
27	spook	45	80	20	96	73	15
28	spyone	60	70	58	70	73	15
29	SRAT	35	80	20	50	60	30
30	pirate	64	80	56	92	40	40
31	Red & Black	45	100	20	97	40	0
32	nightbat	61	90	45	98	73	15

Table 3 shows the result of 32 samples. The score in each column represent the evaluation of the corresponding attribute. Based on marks in the table, the calculation of differentiation degree including:

Step1 The total score and single score are sorted respectively according to ranking;

Step2 Choosing the high score group and the low score group by the proportion of 25% and 25% respectively.

Step3 Computing the differentiation degree by the formula:

$$D=\frac{\left(\overline{X_H}-\overline{X_L}\right)}{w} \tag{5.2}$$

$\overline{X_H}$ indicates the average of high score group.

$\overline{X_L}$ indicates the average of low score group.

w indicates the total number of the sample.

For the score of overall, we get:

$\overline{X_{H_{Total}}}$ =74, $\overline{X_{L_{Total}}}$ =48.375

$$D_{Total}=\frac{\left(\overline{X_{H_{Total}}}-\overline{X_{L_{Total}}}\right)}{w}=\frac{69-40}{100}=0.29 \tag{5.3}$$

For the score of initiation, we get:

$\overline{X_{H_{initiation}}}$ =74, $\overline{X_{L_{initiation}}}$ =48.375

$$D_{initiation} = \frac{\left(\overline{X_{H_{initiation}}} - \overline{X_{L_{initiation}}}\right)}{w} = \frac{94.5 - 63.75}{100} = 0.31 \qquad (5.4)$$

In a similar way, we could get:

$$D_{concealment} = \frac{\left(\overline{X_{H_{concealment}}} - \overline{X_{L_{concealment}}}\right)}{w} = \frac{61.125 - 25.25}{100} = 0.36 \qquad (5.5)$$

$$D_{protect} = \frac{\left(\overline{X_{H_{protect}}} - \overline{X_{L_{protect}}}\right)}{w} = \frac{98.5 - 37.5}{100} = 0.61 \qquad (5.6)$$

$$D_{resource} = \frac{\left(\overline{X_{H_{resource}}} - \overline{X_{L_{resource}}}\right)}{w} = \frac{85.25 - 49.75}{100} = 0.49 \qquad (5.7)$$

$$D_{other} = \frac{\left(\overline{X_{H_{other}}} - \overline{X_{L_{other}}}\right)}{w} = \frac{71.714 - 12.5}{100} = 0.59 \qquad (5.8)$$

In general, the differentiation degree above 0.4 is optimal. The differentiation degree between 0.3 and 0.39 is good, could be better if modified. The differentiation degree between 0.2 and 0.2 is qualified, still need to be modified. The differentiation degree below 0.19 is poor, the criterion should be eliminated.

Consequently, we could find that the differentiation degree of the model achieved good in single score, and qualified in overall score.

(3) Validity

In order to illustrate the validity of the assessment result, we choose a group of malware which we could configure its function.

Table 4 shows the score of Freerat2.0 with default configuration and Freerat2.0_D with self-delete configuration.

Table 4. The Comparison of a same Malware with Different Configuration

Name	Freerat2.0		Name	Freerat2.0_D	
Overall		49	Overall		62
Initiation	Service loading:FreeRat	80	Initiation	Service loading:FreeRat	80
Concealment	Traffic concealment：0.0 Kb/s Protocol concealment with Port 80	28	Concealment	Traffic concealment：0.0 Kb/s Protocol concealment with Port 80 **Self-delete**	48
Self-protection	Encrypt with ZLIB-Deflate Backup file creation	82	Self-protection	Encrypt with ZLIB-Deflate Backup file creation	82
Resource-consuming	CPU avg: 6.5% Bandwidth avg:0.0 Kb/s Memory avg:10224.3 Kb	46	Resource-consuming	CPU avg: 0.0% Bandwidth avg :0.0 Kb/s Memory avg:5236.2 Kb	73
Other	frequency conversion File cleaned Service unload	90	Other	frequency conversion File cleaned Service unload	90

The results reveal that the trojan horse which could delete itself after initiation get higher mark. And our model could demonstrate the malware with better ability could get higher score.

6. Related Work

The researches on malware detection are hot. And the malware threat assessment plays a more and more significant role in nowadays malware defense and cleans up. However, there aren’t many literatures on the topic of malware threat assessment. This paper will discuss the state of the art of malware threat assessment in detail.

6.1. Malware Threat Assessment

Robert J. Bagnall *et al*. designed a malware rating system (MRS) [17]. The system focuses on three aspects of malware: 1. potential threats of payload, which characterize the potential threats of the code against targets' degrade and damage; 2. potential threats of proliferation, which depicts the speed and convenience of malware proliferation; 3. hostility level, which portrayed payload's malicious intent. These elements are assigned with different weights combined expertise to calculate the initial category rating by mathematical formulas. Then the system combines with the specific environmental applications of MRS to adjust the initial rating through *r* factor to portray the malware threats to different organizations. The shortcomings of that system including: a. the weights assigned to the 3 aspect of malware mentioned above is totally depend on the expertise; b. the third aspect, hostility level, focused on the intent of malware writers which is somewhat subjective and difficult to judge; c. the system couldn't distinguish the threat posed by different type of malware.

Zhang *et al*. proposed a method on malware risk assessment [18]. The method first identifies the infectiousness and destructiveness of malware. The identification of infectiousness based on the number of infected sites, the number of infected computers, the geographical distribution and the industry distribution of infection. The identification of destructiveness based on the propagation, destructive power and complexity of the malware. Then they construct the probability matrix of malicious code security event and the loss matrix of malicious code security event. Based on the probability matrix and the loss matrix, they construct the damage matrix of malicious code to get the risk assessment of malware. Their method is similar with the method of traditional information system risk assessment. But the construction of the three matrixes depends on the expertise heavily, and the assessment didn't focus on the code itself.

Zhang *et al*. proposed a method to evaluate the concealment property of malware[19]. They assign different weight to different implementations. But with regard to many implements of a certain criterion, they chose the highest value. And the weight of different criterion is identical, which could not reflect the fact that the threat of malware depend on its implementation to some extent.

Han proposed threat assessment of malware based on the behavior and features of the malicious code[20], such as characteristics of file structure, strings feature, host behavioral(the process behavior, the behavior of the registry, the file behavior and network behavior). The method of threat assessment is mainly depending on analytic hierarchy process (AHP). AHP is an effect way to quantitate the expertise. But it couldn't reveal the connections between the different implementation of a certain attribute.

In order to pick out the most important samples from a ranking list and to relieve stress on human analysts, Zhen Tang *et al*. propose two different criteria to generate a ranking list to evaluate the importance of each malware sample [21]. Criterion 1: How strongly the events are classified as positives. Criterion 2: How anomalous the events are relative to the labeled samples. They generate two groups of scores by SVM from criterion 1, and nearest neighbor with update from criterion 2. Then they generate the final ranking by the product of scores from criterion 1 and criterion 2.

Compared with method mentioned above, our method's advantages lies in:

- The assessment of malware threat focused on the code itself, which is objective in depicting the threat of malware;
- The information fusion model we proposed could evaluate the malware threat aimed at different characteristics of behavior level, attribute level and decision making level;
- The method could assess different type of malware in a more accurate way.

6.2. Multi-Feature Analysis Framework of Malware

Accurate malware threat assessment originated from the identification of the malware behavior. But the current analysis of the malware behavior always based on solitary behavioral feature, such as system call, control flow graph, data flow graph, opcode sequence and structural characteristics of the binary file *etc*.. If security experts cannot get relevant indication out of technical defects, the system may not characterize the behavior of malware. And single feature is vulnerable to various anti-detection techniques [7][8].In this regard, there are some experts and scholars devoted themselves on the research of multi-feature malware detection .

Wei *et al*. [22] proposed a new network security situational awareness model based on information fusion by considering the characteristic of multi-source information in network security research. The model combines multi-source information from a mass of logs by introducing the improved D-S evidence theory to improve the accuracy of situational awareness.

Kong *et al*. [22] proposed an algorithm to detect obfuscated malware based on boosting multi-level features. Followed by disassembly analysis and static analysis, the algorithm takes into account three dimensions: the distribution sequence of opcode, the function call flow graph and the system call flow graph to summarize and analyze the characteristic of malware family. These dimensions combine the statistic and semantic features to reflect the behavior characteristic. Then the algorithm output the judgment of which malware family it belongs to base on weighted voting for a different feature analysis.

Qin *et al*. [24] proposed a malware detection and recognition algorithm based on two-dimensional behavior characteristics. The algorithm summarized and analyzed the system call sequence and system call flow graph of disassemble code to combined the semantic structure with the code structure to represent the act of malware. Then the method construct call classifier according to system call sequence and call flow graph. And it determine the weight depend on the output of the classifier. Last it synthesis the advantages of integrated classifier to draw the final decision outcome.

Xin Hu *et al*. [25] proposed a novel system called DUET by exploiting the complementary nature of static and dynamic clustering algorithms and optimally integrating their results. By using the concept of clustering ensemble, DUET combines partitions from individual clustering algorithms into a single consensus partition with better quality and robustness.

Rafiqul Islam *et al*.[26] presented the classification method integrating static and dynamic features into a single test. They placed their focus on the feature of function length frequency(FLF), printable string information(PSI) and dynamic feature vectors.

Most research focus on the determination of malware maliciousness. We call it determination-oriented multi-feature fusion. The essence of the idea is classifier ensemble based on the single classifiers of a certain feature. And each feature contributes different weight to the decision of the maliciousness. However, the relationships between diverse features have not yet been investigated in depth, which is just one of the contributions of this paper

The model we proposed aimed at the assessment of malware threat, which demands analysis in detail. We call it assessment-oriented multi-feature fusion. The essence of the idea is pattern matching. The feature is used to determine the existence of a certain behavior based on the union of implementations in different levels of a certain behavior. The work of assessment is left to the information fusion model we proposed.

7. Discussion and Future Work

Our research on the information fusion model for malware threat assessment takes the relationships between malware attributes into account, which could improve the accuracy and validity of malware threat assessment greatly. The model decomposes the process of malware threats assessment. According to the different characteristics of each layer, the information fusion model select algorithms, avoiding the defect of using single algorithm, which improves the malware threat assessment of adaptability and scalability.

(1) However, there are aspects still many aspects needs to be discussed.
(2) The performance of our model depends on the result of code analysis. Accordingly, the static analysis and real-time monitor will affect the result of threat assessment. In order to ensure the accuracy and validity of the model, it is necessary to improve the performance of each module continually. Especially, malware writers armed their tool with various advanced anti-anti-virus technology. If we could attempt to assess their threat, we must understand their real behavior.
(3) In this paper, we illustrate effectiveness of the model against trojan horse. However, the criterion for distinct malware is different. Therefore, it is still need to figure out the proper criterion against different malware.
(4) The attributes of a certain malware, the weight of different attribute, the weight of a certain implementation for an attribute are all selected depend on expertise. With different background, the selection is different as well.

8. Conclusion

Our information fusion model characterizes the malware threat with a view of malware’s peculiarities. It could help security professionals to allocate the limited human resource and financial resource to the most effective way to defeat malware. In Section 7 we noticed several issues need to study, which is the focal point of our future work.

Acknowledgements

The work presented in this paper is sponsored by the National Natural Science Foundation of China(No.61472447) and the Key Science-Technology Project of Henan Province of China (No.SP10JH13042).

References

[1] I. A. Saeed, A. Selamat, A. M. Selamat and S. B Abdulaziz. “A Survey on Malware and Malware Detection Systems”, analysis, vol. 3, no.10, **(2013)** pp. 13-17.
[2] ThreatGRID Inc., “Unified malware analysis and threat intelligence”, http://www.threatgrid.com., **(2015)**.
[3] FireEye, Inc. An adaptive defense requires Threat Intelligence. https://www.fireeye.com/products/dynamic-threat-intelligence.html., **(2015)**.
[4] Dell SecureWorks, Inc. Malware Code Analysis. http://www.secureworks.com/cyber-threat-intelligence/malware_code_analysis, **(2015)**.
[5] L. Dobrica and E Niemelä, “A survey on software architecture analysis methods”, Software Engineering, IEEE Transactions on, vol.28, no.7, **(2002)**, pp. 638-653.
[6] K. Yakdan, S. Eschweiler and E Gerhards-Padilla. “REcompile: A decompilation framework for static analysis of binaries. In Malicious and Unwanted Software”, The Americas"(MALWARE), 2013 8th International Conference on, **(2013)**, pp. 95-102. IEEE; Fajardo, PR
[7] C. Linn and S Debray, “Obfuscation of executable code to improve resistance to static disassembly”, In Proceedings of the 10th ACM conference on Computer and communications security, **(2003)**, pp. 290-299. ACM; New York, USA.
[8] J. A. Marpaung, M. Sain and H. J Lee, “Survey on malware evasion techniques: State of the art and challenges”, In Advanced Communication Technology (ICACT), 2012 14th International Conference on IEEE, **(2012),** pp. 744-749; PyeongChang, Korea.
[9] A Rountev, O Volgin and M Reddoch, “Control flow analysis for reverse engineering of sequence diagrams”, Rapport Technique, Ohio State University, **(2004)**.
[10] S Alama, I Traoreb and I Sogukpinar, “Annotated Control Flow Graph for Metamorphic Malware Detection”, The Computer Journal, bxu148, (**2014)**.
[11] Z Fu-yong, D Qi and J L Hu, “Unknown Malware Detection Based on IRP”, Journal of South China University of Technology (Natural Science Edition), vol.4, no. 005, **(2011)**.

Copyright ⓒ 2016 SERSC

[12] Z FuYong, Q Deyu and H JingLin, "MBMAS: a system for malware behavior monitor and analysis", In Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on IEEE, **(2009)**, pp. 1-4.; Wuhan, China.

[13] K. Rieck, T. Holz, C. Willems and P. Düssel, "Learning and classification of malware behavior", In Detection of Intrusions and Malware and Vulnerability Assessment, **(2008)**, pp. 108-125, Springer Berlin Heidelberg.

[14] R Veeramani and N Rai, "Windows api based malware detection and framework analysis", In International conference on networks and cyber security, vol. 25, **(2012)**, Kuala Lumpur, Malaysia.

[15] M Nitin., J Oberheide., J. Andersen, Z. M. Mao, F. Jahanian and J Nazario, "Automated classification and analysis of internet malware", In Recent advances in intrusion detection, **(2007)**, pp. 178-197. Springer Berlin Heidelberg Cheetah Mobile Inc.. https://fireeye.ijinshan.com/ (2015).

[16] R J. Bagnall and G French, "The Malware Rating System (MRS)^{TM}", In Proceedings of the Sixth International Command and Control Research and Technology Symposium (6th ICCRTS). **(2011)**, Québec City, Canada.

[17] J Zhang, X Shu, Z Du, P Cao, S Su and J Wang, "Research on a Method of Malware Risk Assessment", Information Network Security, vol.10, **(2009)**, pp.7-9.

[18] R Zhang, Y Hu and K Zheng, "Method for Evaluating Concealment of Malicious Code Based on Behavior Analysis", Proceedings of Research on the New Progress of Information Communication in China 2013, **(2014)**.

[19] Y Han, "A Research of Malware Detection and Evaluation based on Behavior Analysis", Beijing Jiaotong University, **(2014).**

[20] Z Tang and J Schneider, "Analysis of Prioritizing Malware." http://www.ml.cmu.edu/research/dap-papers/tang.pdf, **(2015)**.

[21] Y Wei, Y Lian and D Feng, "A Network Security Situational Awareness Model Based on Information Fusion." Journal Of Computer Research and Development, vol.3, **(2009)**, pp.353-362.

[22] D Kong, X Tan and H Xi, "Obfuscated Malware Detection Based on Boosting Multilevel Features", Journal Of Software, vol. 22, no. 3, **(2011)**, pp. 522-533.

[23] J Qin, H Zhang and Z Su, "A Malicious Code Detection Method Based on Two-dimensional Behavior Characterization", Computer Technology and Development, vol.6, no. 039**, (2013)**.

[24] X. Hu and K Shin, "DUET: integration of dynamic and static analyses for malware clustering with cluster ensembles", In Proceedings of the 29th Annual Computer Security Applications Conference **(2013)**, pp. 79-88. ACM New Orleans, Louisiana, USA.

[25] R. Islam, R. Tian, L. M. Batten and S Versteeg, "Classification of malware based on integrated static and dynamic features", Journal of Network and Computer Applications, vol. 36, no.2, **(2013)**, pp. 646-656.