ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 1392 -1397
doi: 10.18178/wcse.2017.06.241

A Virus Detection Model Based on Artificial immunity System

Huakang Xing [1], Zhengmin Li [2, 3+] and Zhengping Jin [1]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2] National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC)

[3] Institute of Information Engineering, Chinese Academy of Sciences (IIE, CAS)

Abstract. Many new research of virus detection appeared with the development of machine learning and artificial intelligence. The biggest drawback of traditional virus detection methods is that it's powerless for unknown viruses and new varieties detection. This situation is gradually improved owing to the introduction of intelligent learning algorithm. Anyway, the existing virus detection methods still have some problems such as low detection rate and poor ability of anti-obfuscation. In order to solve these problems, this paper proposed a virus detection model based on artificial immune system and sub-graph pattern mining. First of all, we extract the semantic features of malicious files by unpacking process, static analysis and call graph construction. Then, convert it to DFS code Graph and carry on pattern sub-graph mining to get virus candidate feature library. Finally, use the negative selection algorithm to deal with the virus candidate feature library of malicious files and the feature database of benign files to get virus feature library. The experimental results show that the model has a high ACC value for unknown viruses and new varieties, with an average of over 97%.

Keywords: Virus detection, Artificial immunity system, Negative selection, Semantic features, Sub-graph pattern mining

1. Introduction

With more business activities carried on the Internet, the network security needs more protection urgently. As the most destructive one in the Internet world, computer virus is becoming more and more intelligent and explosive under the driving of economic reasons. The traditional virus detection methods based on signature comparison is weak in detecting of new unknown viruses. People try to study a more intelligent detection method to deal with this challenge.

There are many outstanding research achievements which take advantage of the main idea of machine learning and artificial intelligence technology, represent the cutting edge in the field of virus detection. Wang [1] proposed a signature extraction method of computer virus based on artificial immune system and code relevance in 2011. Because it used the traditional byte code as identifier of virus, its ability of detecting virus variants and anti-obfuscation reduced greatly. Li [2] present a string matching processing unit (SMPU) for virus semantics with regular expression in 2012, which get a good outperforms in patterns match. In 2013, Santos et al[3] extracts sequence of operation code as grammatical features, and contrasts the effect of different virus methods which used statistics, KNN algorithm, SVM, Naïve Bayes and other data mining algorithms respectively. His work provided much reference and guidance for later researchers. Suha [4] proposed a virus detection method based on clonal selection algorithm in the same year. In 2016, YuJie et al.[5] improved the KNN algorithm and proposed a method of virus detection based on sequence mining, and

+ Corresponding author. Tel.: + 15600931143; fax: + 86-010-62283192.
E-mail address: 870810588@qq.com.

achieved good accuracy detection. Then, Aya Hellal et al [6] proposed a graph pattern mining algorithm called MCFSM which achieved a good true positive while maintaining a low false positive partially. But the drawback of this approach is that it's utility dependent on training sets and fluctuated greatly. In addition, some achievements have been made in the detection of intelligent hardware loophole[7], network virus[8,9] and cell-phone virus detection[10] under a specific platform.

This paper proposed a virus detection model based on artificial immune system and sub-graph pattern mining algorithm. It provides a fast and effective virus detection scheme, which ensures high detection rate while maintaining a low false alarm rate. Experiments show that, with the different condition of mixture ratio of malicious files and benign files in the training sets, the model always guarantee a high ACC value of 97% on average. The lowest correct detection rate was 95%, and the highest value reached 98%.

2. Virus Detection Model

Based on the negative selection algorithm and the gSpan algorithm, a virus detection model is proposed, and its technological process is shown in Figure1.

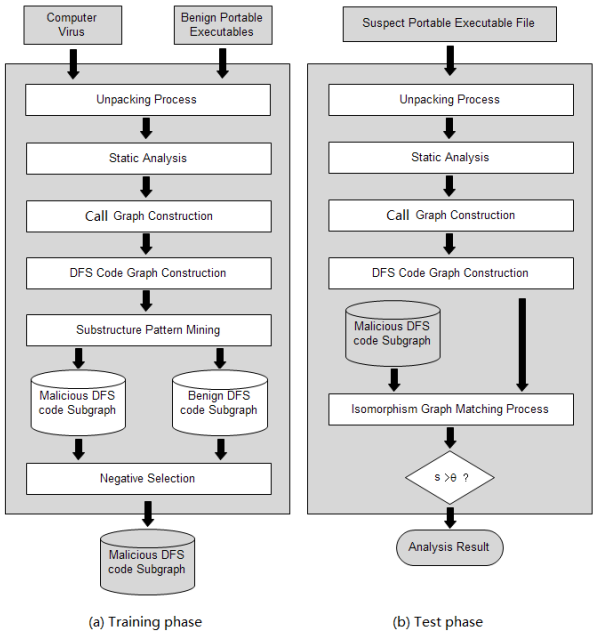

Fig. 1: The architecture of the virus detection model.

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
.text:00401003                push    ecx
.text:00401004                push    ecx
.text:00401005                push    esi
.text:00401006                xor     esi, esi
.text:00401008                call    ds:GetThreadLocale
.text:0040100E                push    7               ; cchData
.text:00401010                lea     ecx, [ebp+LCData]
.text:00401013                push    ecx             ; lpLCData
.text:00401014                push    1004h           ; LCType
.text:00401019                push    eax             ; Locale
.text:0040101A                call    ds:GetLocaleInfoA
.text:00401020                test    eax, eax
.text:00401022                jz      short loc_401043
.text:00401024                mov     cl, [ebp+LCData]
.text:00401027                test    cl, cl
.text:00401029                lea     eax, [ebp+LCData]
.text:0040102C                jz      short loc_401043
```

Fig. 2: Assembly code partition example.

At first, do unpacking operation and static analysis for computer virus and benign file respectively, and get the decompile file as shown in Figure2. Remove the address part of each file and only leaving the operation code and operands with the format processing script. Take its result as input of algorithm1 to construct call graph. Then, convert call code graphs into DFS code graphs in accordance with the definition

of DFS code in [6]. Next, run improved gSpan algorithm to mine isomorphic sub-graph pattern mining to get DFS code sub-graph pattern. Inspired by artificial immunity system, we used negative selection algorithm to deal with characteristic sub-graph pattern set to get final virus detection feature sub-graph pattern library. So far, the training phase has finished. Take suspicious file unpacking, decompile, constructing call graph and constructing DFS code graph. Contrast DFS code graph with virus feature pattern library to get the matching degree, and get the test result.

The prototype of algorithm1 is the transform algorithm in [11]. The main improvements is add internal recursive investigation of procedure call at line 24-28.

Algorithm.1 Transform algorithm

Input: portable Executable Binary File F.

Output: Call Graph $G = (V, E)$.

1. V ← BuildNode(); E ← ∅;
2. $v_i \leftarrow 0$;
3. r ← EntryPoint(F);
4. **while** *r* is not the end of F **do**
5. //The instruction at address r in F.
6. $I_c \leftarrow I[r]$;
7. //The parameter of the instruction.
8. $P_c \leftarrow P[r]$;
9. **if** I_c is System Call **then**
10. **if** v_i is equal to 0 **then**
11. $v_i \leftarrow P_c$;
12. **else**
13. $v_i \leftarrow P_c$;
14. E ← E ∪ Edge(v_i, v_j);
15. $v_i \leftarrow v_j$;
16. **end if**
17. **else if** I_c is *JMP* **then**
18. $v_j \leftarrow GetFirstCall(r)$;
19. E ← E ∪ Edge(v_i, v_j);
20. $v_i \leftarrow 0$;
21. **else if** I_c is *CJMP* **then**
22. $v_j \leftarrow GetFirstCall(r)$;
23. E ← E ∪ Edge(v_i, v_j);
24. **else if** I_c is *Procedure Call* **then**
25. $v_j \leftarrow GetFirstSCALL(r)$;
26. E ← E ∪ Edge(v_i, v_j);
27. recursionDeal(I_c, r);
28. $v_j \leftarrow GetEndSCALL(r)$;
29. **end if**
30. r ← CurrentPorgramCounter(F);
31. **end while**

After dealing with algorithm1, the semantic features of file are represented as behavior call code graph which just as shown in figure3.

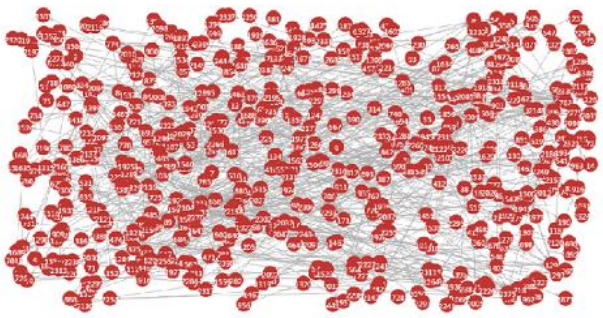

Fig. 3: Examples of call code graph

Algorithm 2 Improved DFS code comparison rules.

Input: $g_1 = (e, e_2, \ldots, e_n)$: the DFS code of graph1 , $g_2 = (e'_1, e'_2, \ldots, e'_m)$: the DFS code of graph2, w: a large constant number.

Output: the relation of g_1 and g_2

1. set k = $\min(n, m)$
2. set s = $\sum_{i=1}^{k} e_i * 10^i$;
3. calculate s_1 *and* s_2;
4. **if** m == n and $s_1 == s_2$
5. return $g_1 = g_2$;
6. **else if** m>n and $s_1 == s_2$
7. return $g_1 < g_2$;
8. **else**
9. return $g_1 > g_2$;

Algorithm2 made some adjustment of gSpan algorithm [12] in the DFS code comparison rules. This improvement makes the time complexity of DFS code comparison decrease from$O(n^2)$ to $O(n)$. The details are just as follows:

Take all process above-mentioned for the malicious file and benign file in training set and get final virus features library. For the suspicious file detection process, we need to carry out the same processing for it to get the DFS code graph. Compare it with virus characteristic pattern library to obtain test result.

3. Datasets and Results

Our experiment got 3259 malware programs from VX heaven (http://www.vxheaven.org/). Beyond that, we collected 810 benign files from the platform of windows. There are two groups of experiments in this paper. All experiments were performed on a computer with windows 7, with a core is i7 CPU of 4*2.0GHZ, and 8GB of memory. All of this two groups experiments were performed on a computer with windows 7, with a core is i7 CPU of 4*2.0GHZ, and 8GB of memory. And the main evaluate measure as follow:

- True Positive (TP): the number of malware correctly recognized
- True negative (TN): the number of benign files correctly recognized
- False positive (FP):the number of benign files recognized as malware
- False negative (FN):the number of malware recognized as benign files
- Detection rate (DR): $TP/(TP+FN)$
- False positive rate (FPR): $FP/(FP+FN)$
- Accuracy (ACC):$(TP+TN)/(TP+TN+FP+FN)$

Table1 shows the values of DR, FPR and ACC at different ratio of training malware and training benign files. This experiment result got with 250 malware and 60 benign files as test sets. The value of DR reached the minimum when the training set has 500 malicious files and 150 benign files. The maximum of DR attained with 2500 malicious files and 500 benign files, which gained 100%.

Table I: DR, FPR and ACC using Different Training Sets

Malware/Benign file	DR	FPR	ACC
500/150	0.97	0.15	0.95
1000/250	0.98	0.08	0.97
1500/350	0.98	0.10	0.97
2500/500	1.00	0.13	0.97
3259/810	0.99	0.05	0.98

Table1 shows the trend of ACC value. Within certain range, the number of malicious software in training set increases, the ACC is also increased. There is a positive association between them.

Table II: DR, FPR and ACC with Different Testing Sets.

Malware variant/Benign file	DR	FPR	ACC

3/33	1.00	0.09	0.92
5/35	1.00	0.06	0.95
7/37	0.86	0.05	0.98
9/39	1.00	0.03	0.98

Table2 shows the result of the second group experiment. Firstly, extracts a number of computer virus file randomly in training sets to confuse. Confused files are treated as a new variant of the malicious file which used as the input for test phase of model. The virus feature library used in this experiment is obtained with condition of training set is composed of 3259 malwares and 810 benign files. Just as above shown, three groups got 100% DR value while only one group is 86%. It means that only one virus variant was failed to be detected successfully. Generally, the model has great ability to anti-obfuscation.

Table III: The DR and ACC with Different Methods.

Methods	AVG(DR)	MAX(DR)	AVG(ACC)	MAX(ACC)
Proposed method	0.98	1.00	0.97	0.98
Hellal et al.(2016)	0.94	1.00	0.89	0.92
Fan et al.(2016)	0.96	0.96	0.94	0.95
Wang et al.(2011)	0.95	1.00	0.95	0.99

As shown in Table3, the virus detection model proposed in this paper is superior to the other three methods on performance of DR. The average value of ACC was higher than the other three schemes while the highest value of ACC was 0.98 which slightly lower than Wang's 0.99 in 2011 unfortunately.

The horizontal axis represents the measures, and the vertical axis indicates the value of items. The defect of maximum of ACC, which more intuitive in figure4, is further to research and optimization.

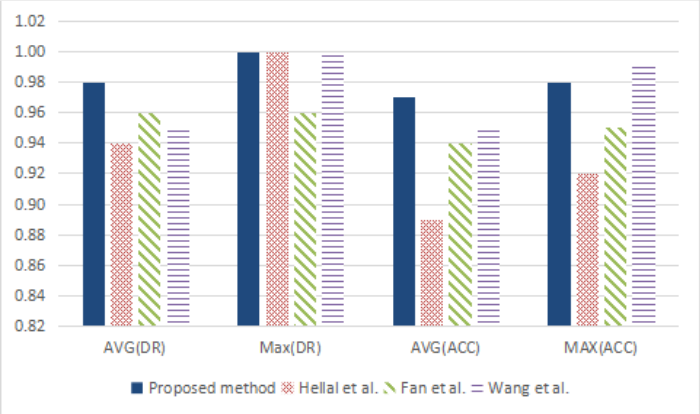

Fig. 4: Comparison of experiment result with different methods

4. Conclusion

This paper has presented a virus detection model which based on the principle of artificial immune system. We proposed a better call code graph construction strategy and a better comparison rule of DFS code. It turns out that our virus detection model has a good anti-obfuscation power. The introduction of artificial immune increased the resolution of virus characteristic patterning database. We found that the performance index including detection rate, false positive rate, accuracy and anti-obfuscation achieve high level.

5. Acknowledgements

This work is supported by NSFC (Grant No. 61502044), and the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

6. References

[1] Wang Wei, Zhang Pengtao, Tan Ying, et al. A method of computer virus feature extraction based on artificial immune and code correlation [J]. Journal of Computer Science, 2011, 34(2):204-215.

[2] Li Q, Li J, Wang J, et al. A pipelined processor architecture for regular expression string matching[J]. Microprocessors & Microsystems, 2012, 36(6):520-526.

[3] Santos I, Brezo F, Ugarte-Pedrero X, et al. Opcode sequences as representation of executables for data-mining-based unknown virus detection[J]. Information Sciences, 2013, 231(9):64-82.

[4] Afaneh S, Zitar R A, Al-Hamami A. Virus detection using clonal selection algorithm with Genetic Algorithm (VDC algorithm)[J]. Applied Soft Computing, 2013, 13(1): 239-246.

[5] Fan Y, Ye Y, Chen L. Malicious sequential pattern mining for automatic virus detection[J]. Expert Systems with Applications An International Journal, 2016, 52(C):16-25.

[6] Hellal A, Romdhane L B. Minimal contrast frequent pattern mining for virus detection[J]. Computers & Security, 2016, 62:19-32.

[7] Fernandes E, Crispo B, Conti M. FM 99.9, Radio Virus: Exploiting FM Radio Broadcasts for Malware Deployment[J]. IEEE Transactions on Information Forensics & Security, 2013, 8(6):1027-1037.

[8] Zhang B, Yin J, Wang S. Research on Virus Detection Technology Based on Ensemble Neural Network and SVM[J]. Neurocomputing, 2014, 137(137):24-33.

[9] Kim D W, Yan P, Zhang J. Detecting fake anti-virus software distribution webpages[J]. Computers & Security, 2015, 49(C):95-106.

[10] Elish K O, Shu X, Yao D, et al. Profiling user-trigger dependence for Android malware detection [J]. Computers & Security, 2015, 49:255-273.

[11] Lee J, Jeong K, Lee H. Detecting metamorphic malwares using code graphs[C]// ACM Symposium on Applied Computing. DBLP, 2010:1970-1977.

[12] Yan X, Han J. gSpan: Graph-Based Substructure Pattern Mining[C]// IEEE International Conference on Data Mining. IEEE Computer Society, 2002:721.