AN APPROACH TO FAST MALWARE CLASSIFICATION BASED ON MALWARE'S META-DATA USING MACHINE LEARNING TECHNIQUE

Pham Van Hung† Toshinori Usui† Kunihiko Shigematsu‡
Keiji Takeda†
†Faculty of Environmental Information, Keio University
‡Graduate School of Media and Governance, Keio University
252-8520, 533 Endo ,Fujisawa-shi,Kanagawa,Japan
Email: {kid,alc,sigematu,keiji}@sfc.wide.ad.jp

Abstract
With the rapid increase of malware, it is important for malware analysis that it is classifying unknown malware files into malware families in order to characterize the type of behavior and static malware characteristic accuracy. In this paper, we introduce an approach to perform fast malware classification based on meta-data of malware's file. We used a machine learning technique called decision tree algorithm to classify malware rapidly and correctly. Experimental results with the malware samples show that our system successfully determined some semantic similarity between malware and showed their inner similarity in behavior and static malware characteristic.

1 Introduction

Malware is the generic name for all type of malicious of malicious software such as Virus, Worm, Spyware, Trojan, and Rookit. According to Eugene Kaspersky, in 2010 the number of malware samples increased by 20 millions [4]. Antivirus or anti-virus software is used to prevent, detect, and remove malware, including but not limited to computer viruses, computer worm, trojan horses, spyware and adware [3]. To create antivirus software, virus researchers have to analyse malware to find common areas that all viruses in a family share uniquely, and can thus create a set of data in order to detect viruses. It is difficult to analyse malware with a vast amount of sample so that it is necessary to have an automatically malware classification system. In this paper, we introduce an approach to perform fast malware classification based on malware's meta-data using machine learning technique, known as decision tree.
With dynamic malware analysis technique, researcher execute malware in the Virtual Machine and use ProcMon, RegShot, and other tools to detect what registry malware changed and file malware created. Unfortunately, these dynamic techniques are susceptible to a variety of anti-monitoring defenses, as well as *time bombs* or *logic bombs* and can be slow and tedious to identify and disable code analysis techniques to unpack the code for examination [15]. Further more, it takes large amount of time to prepare environment to analyze malware such as virtual machine environment but some malware can not be executed in virtual machine environment.
With the static malware analysis technique, researcher perform reverse engineering using IDA Pro and Ollydbg tool to analyse malware by seeing the structure of malware, in order to discover its purpose and functionality but it takes a lot of time to see the malware structure.
In this paper, we introduce the approach to perform fast malware classification based on meta-data of malware's file, using machine learning technique, known as decision tree algorithm.
We will focus our efforts as follows:

- Automatically perform fast malware classification based on malware file's meta-data using a machine learning technique, called decision tree algorithm to classify unknown malwares or subspecies rapidly and correctly.
- Help researcher know which family malware belongs to and detect some semantic information about malware so they can make.

2 Related work

Another approach to malware clustering is the use of n-grams Signatures as known from natural language processing [11]. Calculating these signatures is linear in the sample size with a feasible linear factor c, but since n-grams do not define a distance metric, this approach cannot solve in a speed faster than $O(n^2)$. In this paper, our goal is fast malware classification based on meta-data of malware's file and our approach accomplish malware classification with a complexity of $0(n \log n)$.
A totally different approach is automatic examination of inter-procedural call graph and the intra-procedural flow graphs of malware[7]. Using graph matching, similarity in the code to compare the similarity between two malwares. But it takes a lot of time to calculate the inter-procedural call tree and intra-procedural flow graph and compare the similiarity between two call trees in order to calculate the distance between two malwares. Further more, needs to unpack the sample if it was packed with an executable packer.

To achieve fast malware classification, we cluster malware by famous malware family names which have some semantic information, and use the clustering malware as the training data. Our malware classification system classify unknown malware into malware families rapidly and correctly, and determined their inner similarity in behavior and static malware characteristic.

3 The system architecture

Our system use machine learning technique, known as decision tree algorithm for malware classification system. Our goal is fast malware classification and we use decision tree algorithm to achieve that goal. Classification is a form of supervised learning, which requires training data, with known input/output, to form priori knowledge [12]. As shown in Figure 1, Our system contains three parts.

- First part: Read binary file to take meta data and input to database.
- Second part: Manually cluster malware which load from database.
- Third part: Use decision tree algorithm to classify malware.

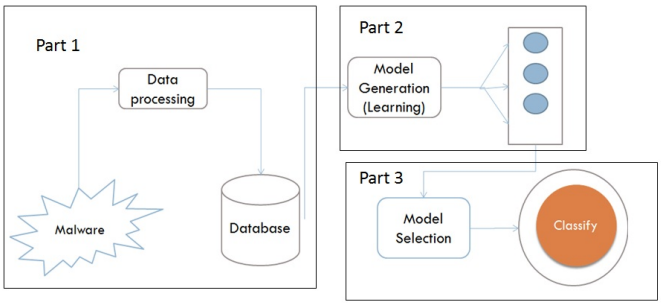

Figure 1. The system architecture.

We want to analyse a vast amount of malware so our system use database to store malware file's meta-data, to easily control a vast mount of data and easily share in the internet by web browser. The flow of data is shown in the Figure 1. We read binary file meta-data and input all of the meta-data in to database. In the second part, our system export meta-data in database as the training data to create the decision tree for classifying malware. Lastly, we use the decision tree algorithm which created in the second part to classify unknown malware into the different malware families.

4 Classification based on machine learning technique

4.1 PE file meta-data

PE is designed as a common file format for MS-DOS. As shown in Figure 2 PE header consists of four parts, a MS-DOS section, PE header, a section table, images pages.

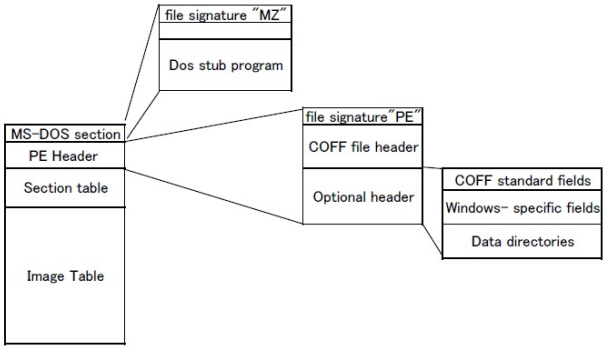

Figure 2. Layout a file in PE header format.

MS-DOS section start with the MS-DOS file signature for executable("MZ"). At the file offset 0xD8, The PE header starts with two characters "PE" as know as the PE header file signature.
The file header contains seven fields shown in Figure 3. The number of the members in section is determined by *NumberOfSection* file in file header.

Member	Offset	Size
Machine	000000DC	Word
NumberOfSections	000000DE	Word
TimeDateStamp	000000E0	Dword
PointerToSymbolTable	000000E4	Dword
NumberOfSymbols	000000E8	Dword
SizeOfOptionalHeader	000000EC	Word
Characteristics	000000EE	Word

Figure 3. Layout of a file header

The following of the file header is the Optional header which consists of three parts. The first part is the COFF standard fields which contains the sizes of various parts of code and the *AddressOfEntryPoint* which indicates the location of the entry point of the application and, the location of the end of the Import Address Table(IAT).
In order to create a fast malware classification system, we only use PE header's meta-data to classify malware. PE header includes meta-data of MS-DOS section, COFF file header, Optional header, and Section header.
There is a problem that PE file's meta data value is so large

that it is very difficult to detect unknown malware which have meta-data's different to the training data of our system. We index meta-data of PE header file which have semantic information for malware classification. For example, normally *ImageBase* field of PE header file have value 400000h [16] so that the vaue of ImageBase of malware file in our system is 0 if it has 40000h and it is 1 if it has other value, and in this research we call that value is semantic value. In this approach, all of PE file's meta-data is used. For example, consist of *HeaderFilesize*, *AddressOfEntryPoint*, *Sizeofsection*, *Imagebase*, *Numberofsection*, *StackCommit*, *SizeHeap Commit*, *SizeImage*, *Characteristics*. We calculated the semantic value of all malware file's meta-data.

4.2 Clustering

Virustotal is a service that analyzes malware and facilitates quick collection of viruses, worms, trojans, and all kinds of malware detected by antivirus vendor such as Norton and Kaspersky [2]. In this research, we use virustotal service to take the name of malware provided by Kaspersky and use it in order to cluster malware into malware families which have semantic information. Malware families is shown in table:
As shown in Figure 5, We use http post technique to automatically get name from virus total service and cluster malware into malware families which shown in Figure 4.

Family name	Define
Worm AutoRun	"A program that secretly and maliciously integrates itself into program or data files. It spreads by integrating itself into more files each time the host program is run" [1]
Trojan Agent	"A trojan, or trojan horse, is a seemingly legitimate program which secretly performs other, usually malicious, functions. It is usually user-initiated and does not replicate" [1]
Keylogger Win32.Zbot	"The primary payload of Trojan:W32/Zbot variants focuses on stealing online banking information. They also have limited backdoor and proxy capabilities" [1]
Trojan PSW Magania	"This type of trojan steals passwords and other sensitive information. It may also secretly install other malicious programs" [1]
Trojan Downloader	"This type of trojan secretly downloads malicious files from a remote server, then installs and executes the files" [1]
Trojan OnLineGames	"This type of trojan steals passwords and other sensitive information. It may also secretly install other malicious programs" [1]

Figure 4. List malware family in our system

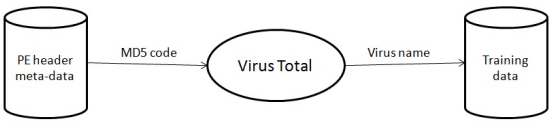

Figure 5. Clustering method.

4.3 Classification

To make malware classification rapid and correct, we use decision tree algorithm. We make the malware classification which easy to update a new malware family, our classification use decision tree algorithm to determine each malware family. For example, based on training data taken from clustering part, we create six decision trees. In the worm autorun decision tree, shown in Figure 6.
We use malware' meta-data as input data for each decision tree, and the the decision tree determine that the malware belongs to worm autorun family or not. If the input malware belong to worm autorun then malware classification detect the family that malware belong to, else we continue to the next decision tree. We use list malware PE header file's meta-data :Magic, MajorLinkerVersion, MinorLinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion, MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, Reserved1, SizeOfImage, SizeOfHeaders, CheckSum, Subsystem, DllCharacteristics, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, LoaderFlags, NumberOfRvaAndSizes, Machine, NumberOfSections, TimeDateStamp, PointerToSymbolTable, NumberOfSymbols, SizeOfOptionalHeader, Characteristics. A value of each field in PE header have semantic for decision tree algorithm if it is appeared at 10% in malware training data. With this approach, we created the list semantic value of meta-data of malware. For example, this is list meta-data we created :1, 2, 6, 4, 0, 1, 1, 4, 0, 200, 4, 0, 0, 0, 4, 0, 0, 1.00E+00, 200, 0, 2, 0, 100000, 4000, 100000, 1000, 0, 10, 14c, 2, 1000, 8, 0, e0, 3, 818e. This is value before we calculated :35328, 10b, 2, 0, 6200, 0, 200, b32e, b000, 9000, 400000, 1000, 200, 3, 0, 0, 0, 4, 0, 0, d000, 400, afe8, 2, 0, 1000, 1000, 10000, 0, 0, 10, 14c,

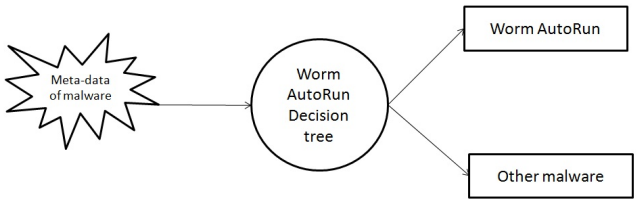

Figure 6. Worm autorun decision tree.

6, 0, 0, 0, e0, 10e. Our malware classification part is shown in Figure 7. We use a training data taken from the malware clustering part, in order to create the order of decision tree that make malware classification system correct. The decision tree we created, shown in Figure 8. When we use PE header's meta-data of malware to input into the worn autorun decision tree to determine unknown malware. We determine it belongs to worm auto run family or not.

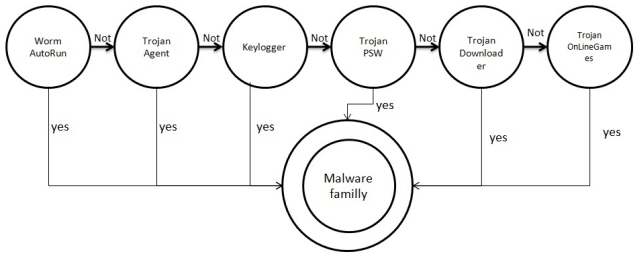

Figure 7. Malware classification system.

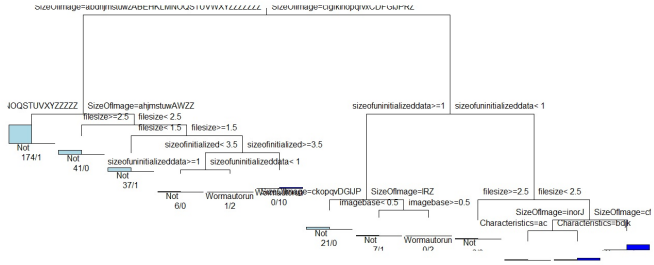

Figure 8. Worm autorun decision tree.

5 Experimental result and analysis

5.1 Data collection

As stated previously, we obtained 935 malwares, taken from our honeypot system, in order to make experimental result of our malware classification system. The malware PE file consisted of backdoors, worm, Trojan, rookit, and packer. We cluster them into six malware families shown the the Table 5. The number of malwares in each family shown in this table.

Family	Number
Worm Autorun	12
Trojan Agent	59
Keylogger	72
Trojan PSW	87
Trojan Downloader	264
Trojan Online games	82
Other family	359

Figure 9. Experimental result table

5.2 Experimental result and analysis

As previously stated, we obtained 935 malware PE file, used 635 malware PE file for make six decision tree. Therefore, we used 200 malware meta-data as training data, in order to sort six decision tree, and we make the best order of decision tree which shown in Figure 10, this order have the best experimental result with 200 malware training data. Lastly 100 malware meta-data to test experimental result our system. With experimental result, the order of decision tree is shown in Figure 10.

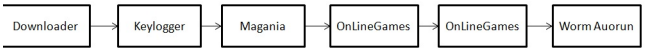

Figure 10. The best decision tree order.

With the order of decision tree, we use 100 malware to test the experimental result our system. In Table 11, we show the experimental result of our system. Some of number in axis show these total number of malware in that family, and that number is separated into group that these malware has been classified by our system. For example, we have 4 malware in trojan agent family, and we successfully classify 3 malware into trojan agent family and have false to classify 1 malware into other family. Therefore, Our system also recognizes the trojan agent family with 75 % accuracy.
W: Worm Autorun, TA: Trojan Agent
K: Keylogger, TP: Trojan PSW
TD Trojan Downloader
TO: Trojan Online games
O: Other family
Our system is useful to help virus researcher determined the malware family that unknown malware belongs to. The malware family contains Worm Autorun, Trojan Agent,

Malware	TA	K	TP	TD	TO	W	O	Accuracy
TA	3	0	0	0	0	0	1	75%
K	0	2	0	0	0	0	2	50%
TP	0	0	2	0	1	0	0	66%
TD	1	0	0	2	8	0	3	53%
TO	0	0	0	0	0	0	0	0%
W	0	0	0	0	0	1	1	50%
O	4	4	2	3	15	2	41	57%

Figure 11. Experimental result

Keylogger, Trojan PSW, Trojan Downloader,Trojan Online games, known as famous malware family. Virus research who know the family of malware can easily determined some semantic similarity between malware and showed their inner similarity in behavior and static malware characteristic.

6 Conclusion

A vast amount of new malware sample each day is the difficult problem in malware analysis. Our malware fast malware classification system successfully classify unknown malware into malware family which have semantic characteristic. We strongly believe that our system is useful for malware analysis to determine malware behavior and semantic malware characteristic to more easily analyse a vast amount of malware. However, there is a problem that our system use only malware meta-data and can not detect the malware family which have same program structure.
Surprisingly, our system cannot be used to classify W32 malware which does not have the signature of PE file signature.

6.1 Future work

Our system successfully classified unknown malware into malware family. In the future work, we use Windows API in order to make our system determined unknown malware into the malware families, which have similarity of program structure.

References

[1] *http://www.f-secure.com*. 14 Jun 2011.

[2] *http://www.virustotal.com/*. 14 Jun 2011.

[3] *http://en.wikipedia.org/wiki/Antivirus-software*. 14 Jun 2011.

[4] *http://www.kaspersky.co.in/news?id=207576357*. 14 Jun 2011.

[5] *http://www.csn.ul.ie/ caolan/pub/winresdump/ winresdump/doc/pefile.html*. 24 Aug 2011.

[6] *http://www.viruslistjp.com/analysis/?pubid=204792101*. 17 Feb 2010.

[7] Silvio CESARE, *Fast Automated Unpacking and Classificationof Malware*. Masters Thesis,Central Queensland University, 2010.

[8] Jingjing Yao, Qiang Hou, *Malicious executables classification based on behavioral factor analysis*. International Conference on e-Education, e-Business, e-Management and e-Learning, 2010.

[9] Yuhei Kawakoya, Makoto Iwamura, Mitsutaka Itoh, *Analyzing Malware with Stealth Debugger*. Information Processing Society of Japan, 2008.

[10] Kevin Coogan, Saumya Debray, Tasneem Kaochar, Gregg Townsend, *Automatic Static Unpacking of Malware Binaries*. 16th Working Conference on Reverse Engineering, 2009.

[11] Tony Abou-Assaleh, Nick Cercone, Vlado Keselj, Ray Sweidan , *N-gram-based Detection of New Malicious Code*. Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004 .

[12] Tony Lee, Jigar J. Mody, *Behavioral Classification*. Presented at the EICAR Conference, May 2006.

[13] Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye, *IMDS: Intelligent Malware Detection System*. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007.

[14] Robin Sharp, *An Introduction to Malware*. Spring 2009.

[15] Georg Wicherski, *peHash: A Novel Approach to Fast Malware Clustering*. December 7, 2008.

[16] goppit, *PORTABLE EXECUTABLE FILE FORMAT*. 2011.