Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 11, November -2017

A Study on Malware Profiling and Result Visualization Design Framework

Bomin Choi[1],Dae-Hoon Yoo[2], Hong-Koo Kang[3], Jun-Hyung Park[4]

KISA(Korea Internet & Security Agency)

Abstract —*Advanced persistent threats increase significantly every year, and the scope of those attacks is expanding from a simple cyber area to national core infrastructures (e.g., communication facilities, power plant, etc.) and industries. Those attacks are mostly caused by the malicious code of the same attacker, and similar types/variants are then distributed continuously. Hence, this paper proposes a framework for malicious code profiling design and result visualization. The framework can classify a large amount of incoming malicious code into similar type groups with common properties and identify the correlation among those data through visualization, in order to effectively respond to cyber breach incidents quickly.*

Keywords- *APT attack, Variants Detection, Malware Profiling*

I. INTRODUCTION

As the overall social infrastructure of Korea increasingly depends on the ICT, the scope of vulnerabilities that can exploit such situation becomes enormous. In particular, the present condition is that the advanced persistent threat (APT) is increasing on a daily basis. According to the Gartner report (2016), the number of APTs increased by 38% in 2015; 60% of them were based on unknown attack methods[1]. That is, the risk and damages caused by APT attacks grow every day, which requires concerned authorities to prepare response by identifying attacks quickly.

Malware variants, which modified some source code of the existing malware, were used for most of the attacks made by the APT organization. The reason is that the cost of new malware development is relatively high, and modifying the existing malware is easier than new development [2]. Furthermore, an environment is available that enables non-experts to easily create and diffuse the malware equipped with various functions massively, as the tools that automatically produce malware appears lately. The number of malware that were reported by 2017 amounted to about 700 million types, and only 70 million malwares are new, according to AV-Test statistics. That is, we can assume that most of distributed malware is the variant malware that modifies a part of the existing original code.

An automated system of identifying similar types/variants is needed to cope with cyberattacks quickly and efficiently, which are caused by such variants. Therefore, this paper proposes a framework for malware profiling and result visualization design that can manage the profile of each malware systematically and support the estimation of the malware created by the same attacker/maker by classifying similar type malwares having common characteristics.

II. BACKGROUND

World-leading cyber security certification organizations such as FireEye, Kaspersky, and Symantec use the similarity of various information extracted from the malwares of major APT attack cases (e.g., 3.20, 6.25, Sony) as a breach of security indicator. Those organizations estimate the attack of the same group by identifying the similarity/variant relationship among malware using the file structure, compilation time, and C&C IP/DNS, which are extracted from the codes [3]. Therefore, an effective tool is needed to generate individual profiles for information, which becomes the major interest of analysts, and then classify those profiles into similar patterns in order to identify the relationship among malware quickly. Hence, the next chapter analyzes the characteristics of various information that can be more effective in identifying malware with similar patterns, based on existing studies.

2.1. Static Meta Information Research on Malware

Malware static analysis is a method that analyzes the internal code and structure using disassembler, without executing malware. The static analysis information includes the header and structure of PE (Portable Executable), DLL (Dynamic Linking Library), assembly command, compilation information, string, and SSDeep. Using SSDeep to identify similar malware is the recent trend. SSDeep divides a file into several units with a certain size and creates a hash value for each

block. The similarity level of the file to compare with the original file can be checked to some extent. There is a shortcoming of high detection failure possibility, however, if the file size is small or the file binary is changed structurally. In addition, there is a limit that malware cannot be identified accurately, if obfuscation or packing is applied to the code. The limit is common to all static analysis.

2.2. Dynamic Meta Information Research of Malware

Dynamic malware analysis is a methodology that analyzes the execution of a file by monitoring the code flow and state change of the memory, network, and registry, by executing the file directly. API (Application Programming Interface), network connection information, and process monitoring information can be extracted. As the information is extracted by executing malware actually, there is an advantage of supplementing the limit of the static analysis technique on malware that obfuscation or packing is applied. However, the time cost is high because the code should be actually executed, and behavior is difficult to detect if there is a latent period for certain period of time like the backdoor.

III. PROPOSING FRAMEWORKD

This paper proposes a framework for malware profiling design that can provide information to estimate the same attacker or attack group, by identifying similar/variant malwares that have common characteristics when a large amount of malware flows in. This paper also proposes a visualization technique that can interpret relationship identification among malware more effectively.

3.1. Designing a malware profiling system

Figure 1 shows the conceptual diagram of the proposing malware profiling system, which is composed of three modules - profile creation, classification module, and profiling search and visualization. The next chapter describes the details of each module.

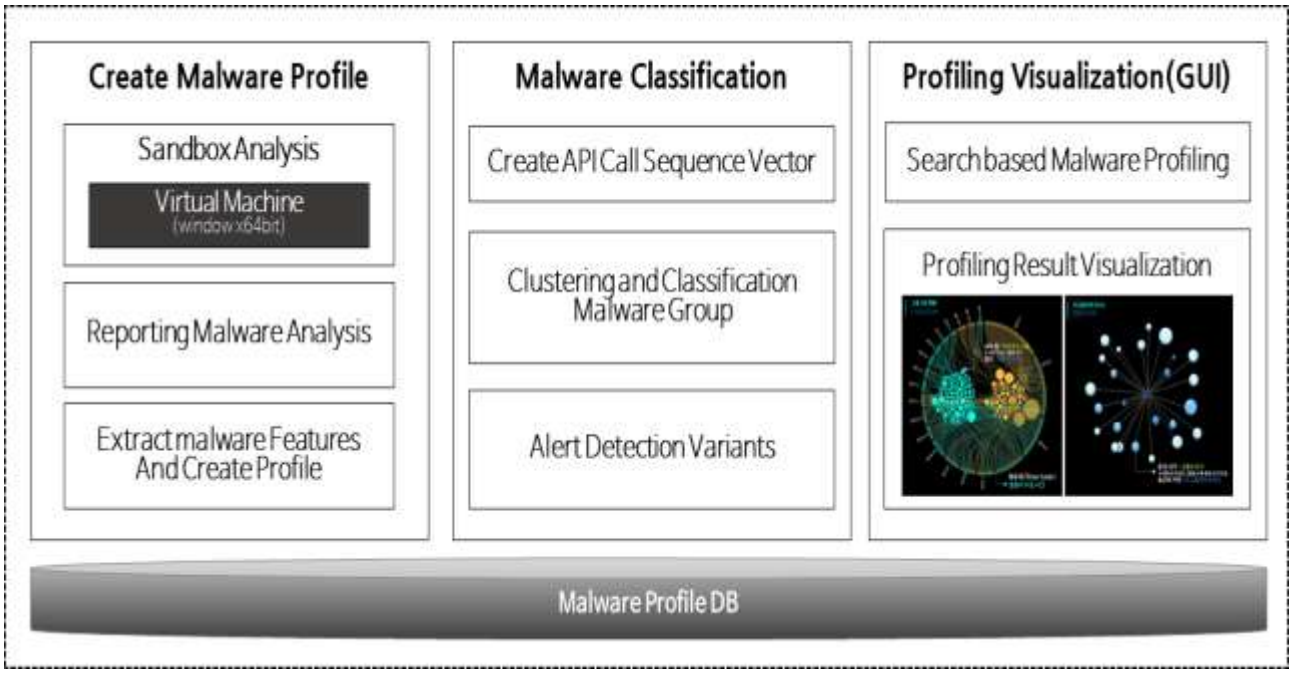

Figure 1. Conceptual diagram of the proposing malware profiling system

3.2. Profile creation module

This module is activated first when malware flows in. This module uploads the pertinent sample onto the sandbox and executes an actual sample and creates an analysis report. The malware analysis report is composed of the meta information, group classification information, static analysis information, dynamic analysis information, network information, and dropped file information.112 types of information corresponding to Table 1 is parsed to create individual profiles and save them in the database.

Table 1. Individual Malware Profile Information

Category	Metadata	Description
basic(14)	sample_id, file_name, ssdeep, …	sample basic info
	group_id, simil_of_center	group classification info
	hash_tag, user_description	user analysis info
static(16)	compile_time	create file time info
	imphash	IAT table info
	.text, .rdata, …	pe section info
	imported_dll_cnt, imports_lib	imports info
	resources_lang/sublang	resource languege info
	certificate_md5, country, …	certificate info
behavior(13)	api_seq, count, call api category, …	call api and behavior info
dropped file(9)	dropped_size, hash, …	dropped file info
network(3)	hosts_ip, country, dns	network info
VT(57)	scan_data, Ahnlab, Kaspersky,…	virus total scan info

3.3. Malware group classification module

This module classifies the group of malware that conducts similar behavior when the profile of individual malware is created. Group classification in this system is conducted on the API call sequence, which is the dynamic analysis information. As mentioned in Chapter 2.1, the reason is that there is a limit on acquiring static analysis information due to packing and code obfuscation.

Groups are classified by applying the 2-gram based cosine sheath formula, using the API call sequence [4]. However, operation quantity can increase geometrically if the original form of the API call is used without modification, because the number of malware increases. The reason is that up to 103,684th (322 types x 322 types) vector can be created if 322 API functions to use are converted to 2-gram, and this type of argument value is inappropriate for applying to clustering and classification logic. The issue was solved by reducing operation cost and converting to a type that is appropriate for using with clustering and classification logic. That is, dimensions are reduced by converting the 2-gram value to 1,204th fixed length vector values using the feature-hashing function before calculating similarity and classifying groups [5].

3.4. Profiling inquiry and visualization module

This module supports the search interface that can retrieve various information of created, saved, and managed malware, and the visualization of the profiling information about the retrieval result. Figure 2 shows the conceptual diagram of the visualization module.

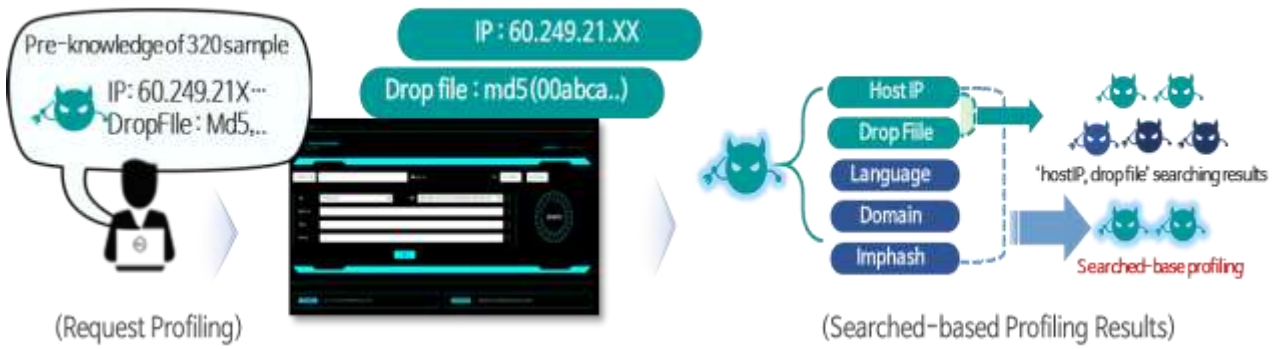

Figure 2. Conceptual diagram of the profiling inquiry and visualization module

The retrieval interface supports search for all information of 112 profile types. Therefore, if users know the part of a certain event, they can specify related malware and search for it. In addition, users can interpret the result more intuitively and conveniently by visualizing Figure 3 and group classification result, similarity relationship among groups, correlation among members retrieved together, and relationship among profiling-retrieved malware.

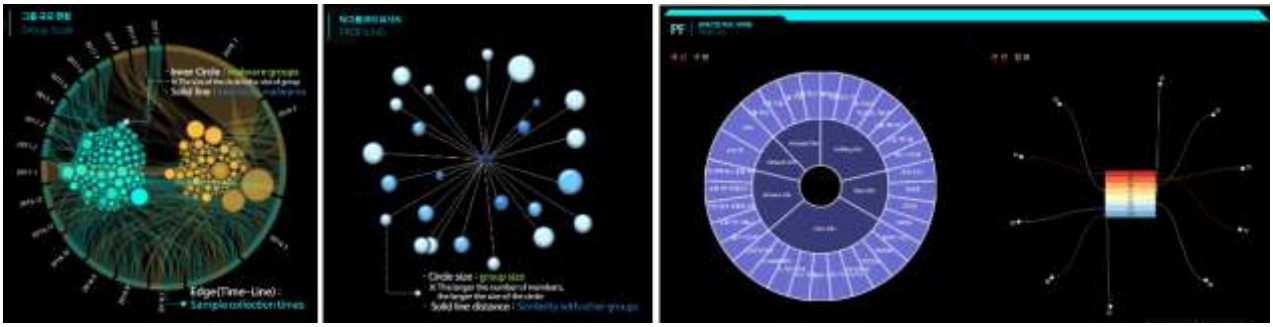

Figure 3. Example of proposing visualization

IV. SYSTEM IMPLEMENTATION RESULTS

If unknown malware penetrates into the system, the analyst can understand and interpret it intuitively by uploading samples onto the system, identifying the malware group that is similar to the pertinent sample, and visualizing the characteristics of those samples. Figure 4 shows the processing overview of this system.

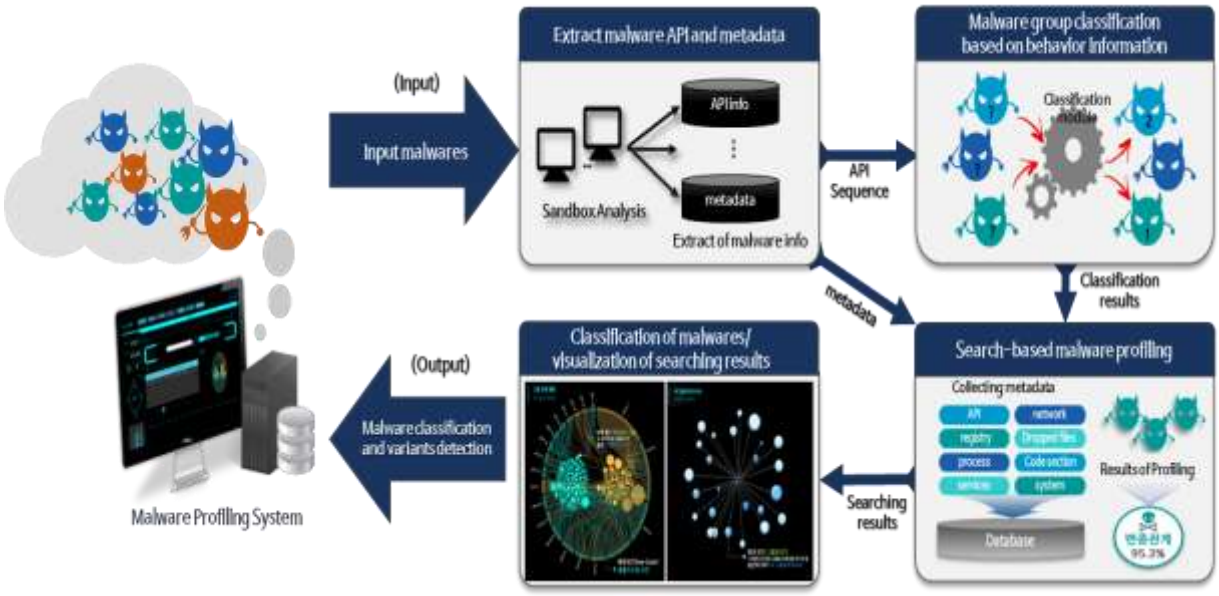

Figure 4. Profiling system overview

Unknown samples can be uploaded onto this proposing system to understand a correlation with other samples through group classification and visualization. Figure 5 shows the pop-up screen of the malware group profile, which is composed of the detailed group information (list of malware classified as the pertinent group, group creation date, group update date, VT information) and the information that can check the reliability of group classification. Analysts can decide response priority faster by presenting the group reliability information in numbers and visualization.

Figure 5. Malware profile pop-up

In particular, each hexagon object on the "similarity map" at the right bottom of Figure indicates each member object included in the group, and the hexagon object in the center becomes the target for comparison with other members. In addition, the color of each hexagon and the distance from the center indicates the similarity of each object. Therefore, if the depth of color is high on the similar map, it means that the pertinent member has a strong influence inside the group, which indicates that similarity to other members is high and classification into the pertinent group is appropriate. On the contrary, if the color depth of the hexagon object is low in the group center, the pertinent sample is not much like other (very similar) members and reliability of classification into the pertinent group is relatively low. That is, it has the virtue of deciding response priority faster and more effectively, using the similarity and reliability information provided by visualization.

V. CONCLUSION

It is difficult for a limited number of analysts to manually select, classify, and analyze entire malware, because more than 1 million malware appear on average each day, due to continuous appearance of variants of malware used for 3.20 and 6.25 cyber terror. Hence, it is believed that an early response system can be developed by detecting major variants using this proposing framework for system design, supporting response to attacks caused by malware (e.g., analysis of related breach incidents), and understanding the attack pattern such as the variant type/target scope of a new breach incident. However, as the type of various data extracted from malware is different, more effective profiling can be achieved by generating an identification signature that can imply those types in an integrated manner and visualizing it. Therefore, more studies seem to be needed that enables users to develop a profiling system that is more concise and easy to analyze, by advancing the data pre-processing function.

ACKNOWLEDGEMNTS

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)
(No. 2016-0-00081, The Development of Integrated Malware Life-cycle Profiling and Attack-group Identification Technology

REFERENCES

[1] V-Test, From https://www.av-test.org/en/statistics/malware/

[2] Hassen Mehadi, et al.(2017, March), "Scalable Function Call Graph-based Malware Classification," In Proc. the ACM on Conference on Data and Application Security and Privacy (CODASPY), pp. 239-248

[3] B' Tupakula.(2013), "On malware characterization and attack classification." Proceedings of the First Australasian Web Conference-Volume 144. Australian Computer Society, Inc.

[4] Liangboonprakong, C., & Sornil, O. (2013, June). Classification of malware families based on n-grams sequential pattern features. In Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on (pp. 777-782). IEEE.

[5] Kang, B., Kim, T., Kwon, H., Choi, Y., & Im, E. G. (2012, October). Malware classification method via binary content comparison. In Proceedings of the 2012 ACM Research in Applied Computation Symposium (pp. 316-321). ACM.

