Featured Articles

Automatic Malware Analysis Technology to Defend against Evolving Targeted Attacks

Hirofumi Nakakoji
Tetsuro Kito
Tomohiro Shigemoto
Naoki Hayashi
Shingo Yamashita

OVERVIEW: As the malware used in targeted attacks has grown more advanced in recent years, the number of cases where existing inbound measures have failed to detect attacks and allowed incursions into the organization has increased. In a situation such as this, it is necessary to clarify the characteristics of the intruding malware so that countermeasures can be taken quickly to prevent the damage from expanding. A dynamic analysis method is used in order to clarify the malware's characteristics, by running the malware in a special analytical environment for behavior observation. Recently, however, types of malware that avoid analysis in analytical environments by restricting execution environments have been growing more common. Malware also exists that attaches to confidential information in a parasitic fashion, and this makes it difficult to simply outsource the malware for external analysis work. In response, the multimodal malware analysis system executes malware under a variety of different analytical environments, so that malware that only runs under a specific environment can still be automatically analyzed. By operating this system in a standalone capacity, it is possible to clarify the characteristics of malware within one's own organization, without the need to rely on external services.

INTRODUCTION

EVER since the world's first computer virus was confirmed in the 1970s, computer administrators have spent the last almost half of a century in confrontation with constantly evolving computer viruses. In the present era, with the wide range of different viruses that exist, any software developed with malicious intent is referred to by the general term "malware," including the computer viruses that generally represent the entire category. As for the targeted attacks and other cyber-attacks that have been making waves recently, malware is being used by increasingly professional criminals as a tool for purposes such as the extraction of money and confidential information, or the destruction of infrastructure systems. To this end, the malware is itself becoming even more advanced, diversified, and sophisticated, and this makes it more difficult to protect against malware using traditional inbound measures such as firewalls and pattern matching.

On the defending side as well, training in targeted attack measures is being conducted among employees as part of a defense strategy against targeted attacks, and operational measures such as improving the security literacy of employees are also being carried out as well as technical solutions. Thanks to the success of these initiatives, information system departments are receiving more reports and samples from employees who have received suspicious e-mail, and this has increased opportunities to acquire unknown malware that existing security measures could not detect, that is, samples that seem to be malware. The information system department then determines whether or not the sample is malware from the perspectives of incident prevention and countermeasures and, if the sample was indeed malware, clarifies the functions of the malware while considering how to respond if the employee's system was infected. It is important to take internal and outbound measures at an early stage in order to prevent damage from occurring or spreading.

This article describes the multimodal malware analysis system that can efficiently clarify the behavior of malware by automating the process that used to be performed manually by malware analysts with advanced and specialized knowledge.

ISSUES IN MALWARE ANALYSIS

Analysis by an expert is necessary to determine if a sample is malware, and what functions the sample possesses as such. A static analysis method employs reverse engineering and other techniques to analyze samples without executing them, while a dynamic analysis method actually executes the samples under a special analytical environment in order to observe their behavior. Although static analysis has the benefit of enabling the detailed clarification of every one of the sample's functions, it is extremely costly because it requires someone with a deep understanding of programs, operating systems (OSs), hardware, and other mechanisms to decipher each individual line of code. Dynamic analysis, on the other hand, can be used to analyze samples that employ obfuscation (code encryption, etc.) without the need to work directly on the samples, and so analysis can be performed relatively quickly in comparison with static analysis. Although dynamic analysis has the advantage of allowing for the confirmation of behavior that is not clarified through static analysis alone (such as behavior after new malware is downloaded from the Internet and executed), it also suffers from a shortcoming whereby the behavior of functions that do not activate themselves during observation cannot be clarified. Usually, when a sample is analyzed, the properties of the sample, the goals of analysis, and the experience of the analyst will be used as a basis to determine how to combine and implement static and dynamic analyses in complementary ways.

Analytical software that supports dynamic analysis has been developed recently, and the open-source software (OSS) Cuckoo Sandbox[(1)] can be downloaded from the website. This type of software employs virtualization technology to safely execute samples within a sandbox (analytical environment), and enable detailed results of observing network communications and application programming interface (API) calls to be acquired, which is why most of the experts who analyze such samples use it in their work. Security vendors are also providing sample behavior analysis services such as ThreatExpert[(2)]* so that the results of analysis can be acquired by submitting samples over the Internet.

As described above, it has become comparatively easier than before for the defending side to analyze samples, thanks to the evolution of technology and tools. Recently, however, the developers of malware have been incorporating mechanisms into the malware they create to avoid detection and analysis, whereby the malware detects the configuration of hardware and software, including the virtual or debugging environment, the version of the OS, installed applications, and so on. The detected information is used by an environment-dependent malware, whose existence has been confirmed, to determine whether or not it is within the environment of its attack target, so that it can change its behaviors accordingly. It has also been confirmed that "downloader" malware exists that downloads secondary malware from a malware distribution server prepared by the attacker, so that the attack can be carried out in stages. There are also malware distribution servers that conceal themselves by checking the Internet Protocol (IP) address of the accessing malware, only distributing secondary malware if the IP address matches that of the target organization, and distributing legitimate content otherwise.

There are many cases where the behavior of malware equipped with this type of mechanism cannot be clarified using existing dynamic analysis software that only works under a specific, previously prepared environment. Also, since malware distribution servers will act as legitimate servers with respect to outsourced external analysis services whose IP addresses do not match that of the organization targeted for attack, the analysts will not be able to clarify the malware's behavior. The existence of malware that acts as parasites in Portable Document Format (PDF) documents and other files that can include confidential information has also been confirmed, and so increasing numbers of companies are hesitant to rely on external services for sample analysis, since the malware is connected to the confidential information. This increases the need for an ability to clarify the characteristics of malware in-house.

In order to resolve these types of issues, the Yokohama Research Laboratory of Hitachi, Ltd. is working on the research and development of technology that can automatically analyze samples under multiple types of analytical environments using dynamic analysis and determine whether or not a sample behaves as malware, as well as the characteristics of the sample.

MULTIMODAL MALWARE ANALYSIS SYSTEM

The multimodal malware analysis system improves the success rate of analysis of environment-dependent malware by employing multiple types of

* ThreatExpert is a trademark of Symantec International Corporation.

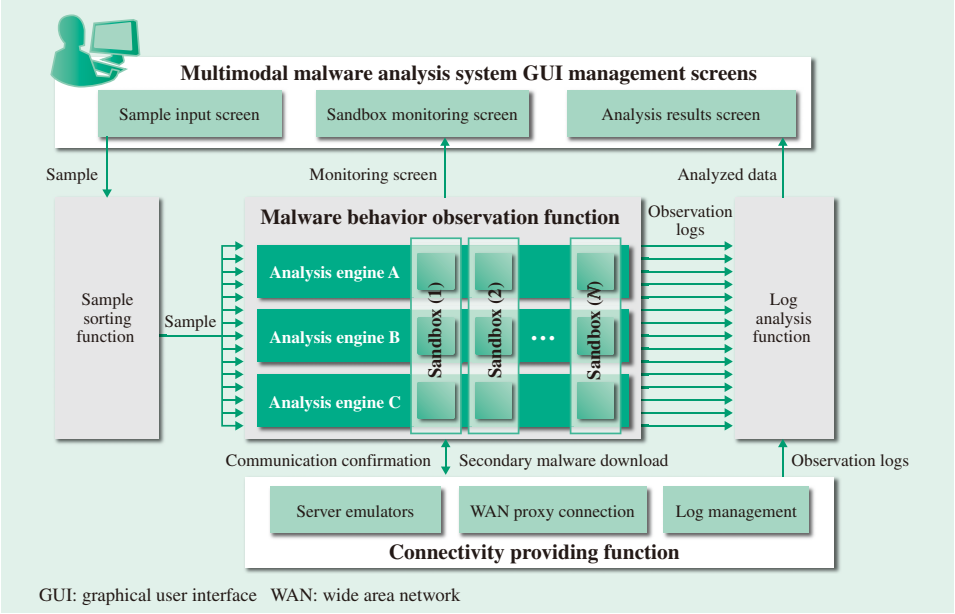

Fig. 1—Multimodal Malware Analysis System Architecture. Multiple types of sandboxes (analytical environments) are prepared to ensure that it is easy for malware to infect and operate, and obtained behavior observation logs are automatically analyzed based on previously acquired analysis know-how in order to create reports. This mechanism automates and speeds up the analytical work previously performed by experts using advanced techniques, while simultaneously achieving a high success rate.

analysis engines and sandboxes during analysis. The architecture of this system is shown in Fig. 1.

When an analyst analyzing a sample that appears to be malware uses the system's sample input screen to input (upload) a sample, that sample is copied by a sample sorting function and simultaneously input into multiple sandboxes configured for use by the malware behavior observation function. Each copy of the input sample is automatically run in the sandboxes, the behavior is observed, and results are output to logs. A log analysis function then gathers the extremely large logs output by the malware behavior observation function (in some cases, millions of lines can be generated for a single sample, amounting to several gigabytes of data), analyzes statistics for the states of activities performed by the samples in each sandbox (file access, registry access, network access, and so on), and extracts files generated by the samples along with any uniform resource locators (URLs) it connects to over the network. Since these processes are automatically run in parallel, the time required for analysis can be greatly reduced, and analysis jobs can be run in overnight batches.

The features of this system are described below.

Malware Behavior Observation Function

The multimodal malware analysis system improves the success rate of environment-dependent malware analysis by analyzing samples in several dozen types of sandboxes. The group of sandboxes is configured using combinations of different analysis engines, hardware, software types and versions, settings, and so on. Although the success rate of environment-dependent malware analysis increases with larger numbers of sandbox variations, since there are limitations in physical machine resources and licenses, preparing every possible combination is not practical.

The sandbox configurations of this system were defined based on the following five selection elements: (1) analysis engine, (2) hardware, (3) architecture, (4) OS, (5) application (see Table 1).

Of these selection elements, three types of analysis engines are used for the multimodal malware analysis system, including the aforementioned Cuckoo Sandbox. Since different virtual machines are supported by different types of analysis engines, the use of multiple types of analysis engines can be expected to be effective not only in terms of analytical performance, but in terms of analyzing malware with virtualization function detection functions as well.

Since there are such an extremely large number of variations, including types and versions of both operating systems and applications, this leads to combinatorial explosion when one considers the various combinations that are possible. This system is designed to infect sandboxes with malware in order to clarify as much of the behavior as possible, and so the selection of environments that are easy for malware to infect and operate in from the perspective of the malware developer (in other words, environments that are most likely to be affected by attacks) is given priority. This is why the OS configurations are designed to differentiate between major operating systems and service packs starting with Windows

TABLE 1. Execution Environment Selection Elements
The five selection elements defined for each execution environment are the analysis engine, hardware, architecture, operating system (OS), and applications.

Analysis engine	Hardware	Architecture	OS	Application
Analysis engine A	• Physical machine • Virtual machine (VMware*1 ESXi)	• 32 bit (×86) • 64 bit (×64)	• Windows*3 XP (SP x) • Windows Vista*3 (SP x) • Windows 7 (SP x)	• Microsoft*3 Office xxxx • Adobe*4 Reader*4 xx • Internet Explorer*3 xx • Adobe Flash*4 Player xxx.x • JRE x.x • Windows Media*3 Player xx
Analysis engine B	• Virtual machine (Oracle*2 VM VirtualBox)			
Analysis engine C	• Virtual machine (VMware Workstation)			

JRE: Java*2 Runtime Environment
*1 VMware is a registered trademark or trademark of VMware, Inc. in the United States and other jurisdictions.
*2 Oracle and Java are registered trademarks of Oracle and/or its affiliates.
*3 Microsoft, Internet Explorer, Windows, Windows Vista, and Windows Media are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
*4 Adobe, Adobe Reader, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

XP, which is reported as being frequently infected by malware. In the application configurations as well, applications with a large number of vulnerabilities (in other words, applications for which vulnerability information has been published a large number of times), are given priority during selection. Information in the JVN iPedia Vulnerability Countermeasure Information Database(3) from January 1, 2012 to August 16, 2013 is used to investigate the number of instances of vulnerability information being published.

Connectivity Providing Function

Recent types of malware are known for using network connection functions to connect to a malware distribution server and download secondary malware, or to connect to a Command and Control (C&C) server in order to receive remote control operations. Also, it has been confirmed that there are some types of malware that attempt to avoid analysis by verifying network communications immediately after infection, to ensure that they have not been copied into an analytical environment.

The multimodal malware analysis system has the connectivity providing functions shown in Fig. 1. This includes functions that emulate servers inside the sandbox in order to respond to various requests from samples directed at major server types including Web servers, File Transfer Protocol (FTP) servers, and Domain Name System (DNS) servers, as well as Wide Area Network (WAN) proxy connection functions (under development) to communicate with malware distribution servers and C&C servers through a proxy connection to the Internet. This allows the behavior of downloader malware to be reproduced with a high degree of accuracy, from when the malware downloads files from specific Web servers through the execution of those files.

Log Analysis Function

The log analysis function identifies the behaviors unique to malware from the extremely large amounts of log data acquired from several dozen different types of sandboxes. The function design (formal knowledge) of the identification algorithms was based on the advanced malware analysis know-how (implicit knowledge) from malware analysis specialists with excellent track records. A number of analytical functions are introduced below:

(1) Determination of the presence or absence of a debugger detection function
(2) Determination of the presence or absence of process injection
(3) Determination of the presence or absence of timed execution
(4) Determination of an external network connection

The behavior detected here often appears as part of the series of illicit activities conducted by malware. For this reason, determining whether or not these behavior patterns are present is a useful method of extrapolating whether or not a sample is actually malware. Inventive techniques based on analytical know-how are also applied as part of each item's determination methods. For instance, during the determination of the existence of an external network connection, multiple types of API calls including minor network connection methods used to avoid detection by malware analysts are monitored, and determination takes a multifaceted approach by analyzing data such as communications traffic and connectivity providing function logs.

Display of Analytical Results

The multimodal malware analysis system has both a function that displays a summary of the operational results of samples in several dozen types of sandboxes, and a function that consolidates and displays a list

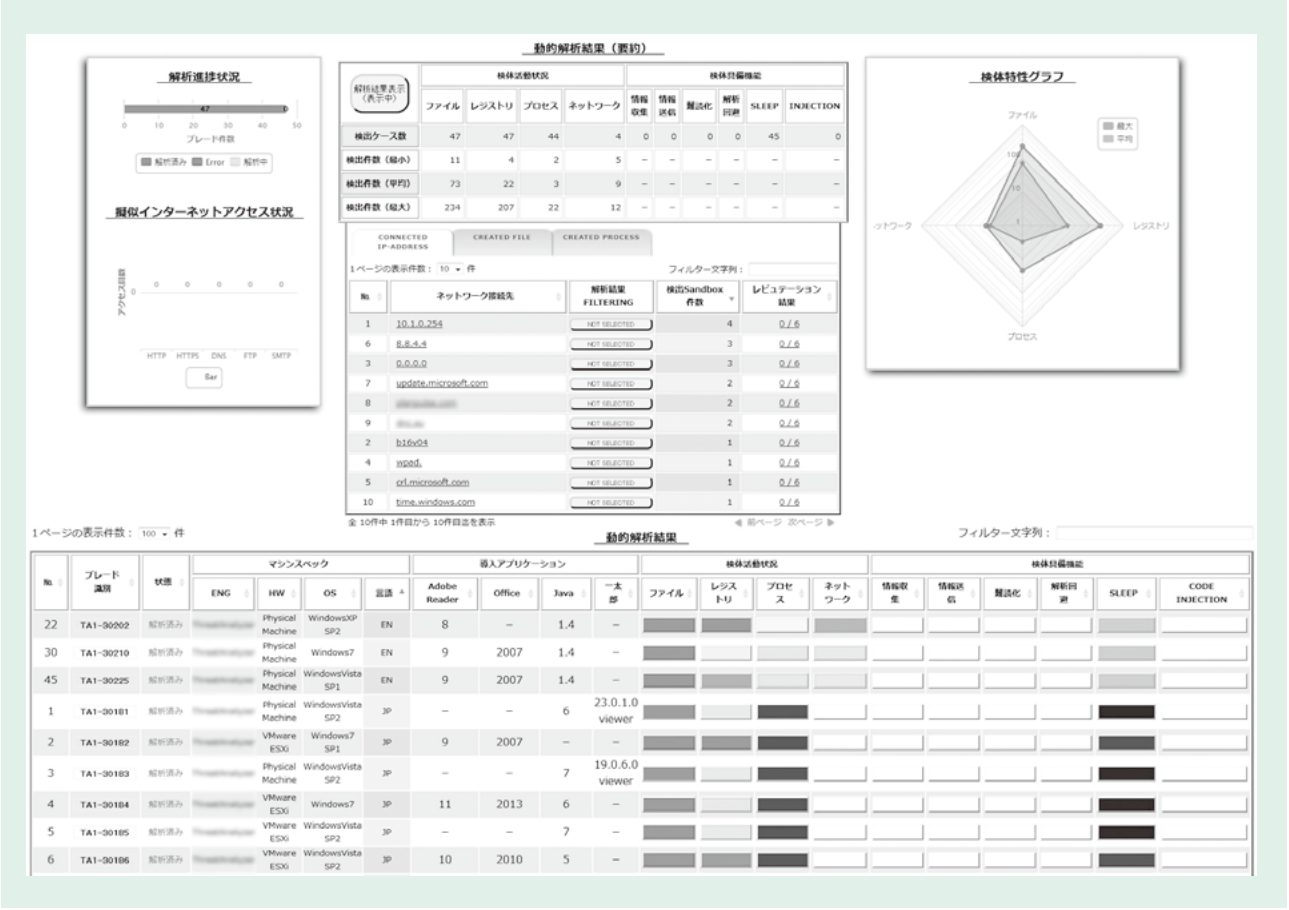

Fig. 2—Sample Analysis Results Screens.
A screen that displays a summary of the malware operational results of several dozen types of sandboxes is provided (top), along with a screen that displays a consolidated list of analytical results for each separate sandbox (bottom).

of analytical results for each separate sandbox (see Fig. 2).

This screen can be used to verify the followings: The URLs connected to by samples; created files; generated process information; the detection results obtained by matching the samples against 16 types of antivirus software pattern files. If a sample is malware, then it is a simple matter to grasp the state of antivirus software support, the URLs that it might connected to over the network if an employee's computer terminal is infected with that malware, any traps placed on employee terminals (malware-related files), and so on. By taking countermeasures such as using this information and a firewall, proxy, or some other method to prohibit communications with the connected URLs, and adding disinfection information to the pattern files of antivirus software, it is possible to utilize defense in depth whereby internal measures and outbound measures are applied if an infection or outbreak occurs in an employee's terminal due to malware slipping past the inbound measures.

MULTIMODAL MALWARE ANALYSIS SYSTEM VERIFICATION

The following is a report of analysis results obtained by using a prototype multimodal malware analysis system in order to analyze several hundred types of samples that appear to be unknown malware, not detectable using the antivirus software of a certain security vendor's antivirus software.

Approximately 80 types of sandboxes were used for this verification work, and it took around 15 minutes per sample to complete the analyses (this was the time required to test a single sample in 80 different sandboxes). Approximately 73% of all the tested samples connected to an external server that appeared to be related to an illegitimate site, and this reconfirms the fact that malware in recent years is characterized by the property of using network connections. Also, by analyzing the malware in sandboxes that reproduce the multiple types of analytical environments that characterize the multimodal malware analysis system,

it was possible to verify the existence of environment-dependent malware that manifests under the following conditions:
(1) Samples that only activate themselves under environments where Microsoft Office 2007/2010 is installed
(2) Samples that only activate themselves under Windows XP
(3) Samples that only activate themselves under a physical environment
(4) Samples that only activate themselves under a physical environment running Windows 7 (except with Service Pack 1)
(5) Samples that do not activate themselves under VMware ESXi or VMware Workstation, but do activate under Oracle VM VirtualBox

In other words, this shows that samples with these properties are difficult to analyze using dynamic analysis under an environment that does not match the proper operating conditions.

Through this verification work, it was confirmed that it is possible to automate malware analysis and clarify the illicit behavior of unknown malware, as well as execute and clarify the behavior of environment-dependent malware. Also, by extracting the number of sandboxes where malware manifests its network connections and other behavior, as well as commonalities in sandbox environment configurations where the behavior manifests, it is possible to derive how easy it is for the malware to manifest, in addition to the environmental conditions under which the environment-dependent malware executes. This type of information can be applied as clues during the construction of analytical environments for use in more detailed analysis.

CONCLUSIONS

This article described the multimodal malware analysis system that automatically clarifies the behavior of suspicious files used in targeted attacks using dynamic analysis, by reporting on the details of the malware's activities.

This system is integrated into a half-rack, all-in-one system designed to operate in a standalone capacity. Since the system can be used as a standalone system for analyzing the behavior of malware, it can analyze downloader malware that only downloads secondary malware from a specific IP address, and samples that are treated as confidential information can be kept within the organization during analysis. By installing this system in the organization's information system department or security operations center, not only is it possible to greatly reduce the cost of malware analysis work performed by experts, this also allows organizations without experts to easily clarify malware threats. This can be expected to have a beneficial effect on defense in depth measures against the latest targeted attacks and other types of cyber-attacks by making it easier to grasp the state of damage, among other benefits.

A safe and secure information technology (IT) environment will be achieved through further reductions in analysis time, expanded log analysis functions, and continued research in automated countermeasures based on information regarding the properties of malware as obtained by this system.

REFERENCES

(1) Claudio "nex" Guarnieri & Cuckoo Sandbox Developers, "Automated Malware Analysis—Cuckoo Sandbox," http://www.cuckoosandbox.org/
(2) ThreatExpert Ltd., "ThreatExpert—Automated Threat Analysis," http://www.threatexpert.com/
(3) JPCERT/CC and IPA, "JVN iPedia—Vulnerability Countermeasure Information Database," http://jvndb.jvn.jp/en/

ABOUT THE AUTHORS

Hirofumi Nakakoji

Enterprise Systems Research Department, Information Service Research Center, Yokohama Research Laboratory, Hitachi, Ltd. He is currently engaged in research and development of measures against cyber-attacks. Mr. Nakakoji is a member of the Information Processing Society of Japan (IPSJ).

Tetsuro Kito

Enterprise Systems Research Department, Information Service Research Center, Yokohama Research Laboratory, Hitachi, Ltd. He is currently engaged in research and development of measures against cyber-attacks. Mr. Kito is a member of the IPSJ.

Tomohiro Shigemoto

Enterprise Systems Research Department, Information Service Research Center, Yokohama Research Laboratory, Hitachi, Ltd. He is currently engaged in research and development of measures against cyber-attacks. Mr. Shigemoto is a member of the IPSJ.

Naoki Hayashi

Enterprise Systems Research Department, Information Service Research Center, Yokohama Research Laboratory, Hitachi, Ltd. He is currently engaged in research and development of measures against cyber-attacks.

Shingo Yamashita

Defense Information Systems Division, Hitachi Advanced Systems Corporation. He is currently engaged in development of information security products.