The Forefront of Cyberattack Countermeasures Focusing on Traces of Attacks

Makoto Iwamura, Yo Kanemoto, Yuma Kurogome, Kazufumi Aoki, Yuhei Kawakoya, Shingo Orihara, and Jun Miyoshi

Abstract

Cyberattacks have become more capable of infiltrating corporate networks with malware by skillfully deceiving the target, and it is becoming increasingly difficult to prevent infections before they occur. Regarding web servers that are open to the public, the frequency of attacks increases and alerts occur more frequently as attack techniques become well known. It is thus becoming difficult to determine which attack to respond to. In this article, the forefront of cyberattack countermeasures focusing on traces left by attacks is discussed to address this issue.

Keywords: cyberattack countermeasures, malware analysis, alert triage

1. Current status of endpoint defense

Targeted attacks aimed at companies and the malware (malicious software) used in those attacks are becoming more sophisticated daily, and it is becoming difficult to prevent intrusions before they occur. Under such circumstances, a technique called endpoint detection and response (EDR) is attracting attention. EDR takes measures to be taken after an intrusion into account under the assumption that an intrusion by malware will be allowed.

Conventional security products prevent infection by detecting the apparent characteristics of malware (such as patterns included in executable files of the malware) as rules before the malware is executed. However, malware that changes its apparent characteristics and escapes detection by security products has started to be used in recent targeted attacks. The apparent characteristics of malware can be changed relatively easily. On the contrary, post-infection behavior is closely related to what the malware is intended to do, and it is considered difficult to change that behavior compared to changing the apparent characteristics. EDR is used to combat such targeted attacks by detecting the behavior of malware after it has begun to spread and leave traces behind.

A rule that detects traces that remain when a computer is infected with malware is called an indicator of compromise (IOC). Depending on the EDR product used, malware infection can be detected with a user-created (“custom”) IOC. Traces left by malware infection and how to generate an IOC to detect them are described in the following sections.

2. Traces of malware infection and their detection

Let us suppose that a file named “mal_a.txt” remains as a trace when malware infects a terminal. To detect that trace, it seems appropriate to prepare an IOC whose file name is “mal_a.txt.” However, when the same malware infects another terminal, if the file name becomes “mal_b.txt,” it cannot be detected by the original IOC. In this case, a little ingenuity is

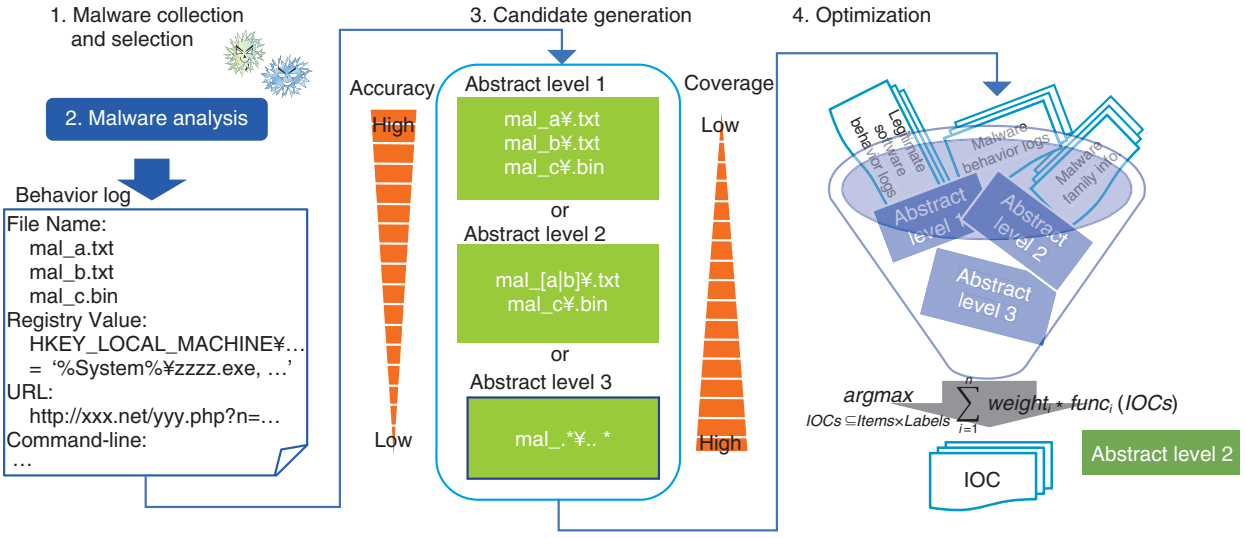


Fig. 1. Automatic generation of IOC.

applied and an IOC is created whose file name is "* .txt" (where * is an arbitrary character string). That IOC will cover both "mal_a.txt" and "mal_b.txt" and might cover other traces that may appear in the future. However, normal applications that are not malware are also running on terminals monitored using EDR. If a normal application creates a file called "leg.txt," the IOC "* .txt" will detect that file as a trace of malware. Although it is harder to change post-infection behavior than the apparent features of malware that prior technology focuses on, an IOC must be expressed in a manner that does not cause false detection while increasing coverage so that it can follow the changes in traces.

One more point must be taken into account when considering what is required of an IOC. Let us suppose an IOC detects that the malware has actually infected a terminal. Much of the subsequent work is left to security engineers, i.e., people. It will be possible to determine, for example, the path the malware entered by, whether it has sent confidential information to the outside, whether any other terminals are infected, and clarify those findings from remaining logs. It may be necessary at times to know what the IOC has detected, improve the IOC, and test other devices. All this is required for an IOC to make it easy for people to see and interpret. Detection criteria are very complex for certain types of machine learning, and some algorithms are difficult to understand, let alone improve. In the field of security, in which people exist in a series of work flows, the interpretability of an IOC also becomes important.

3. Automatic generation of IOC

NTT Secure Platform Laboratories is researching and developing malware-analysis technology that comprehensively identifies the behavior of malware that has various anti-analysis functions. The automatic IOC generation technology [1] introduced here generates an IOC with high detection accuracy, coverage, and interpretability by using the behavior logs extracted with that malware-analysis technology as input. Specifically, an IOC is generated by the following procedure (**Fig. 1**).

(1) Malware collection and selection: Collect and select malware according to the environment to be monitored by the IOC.
(2) Extraction of a behavior log with malware-analysis technology: Analyze malware in a virtual environment dedicated to malware analysis and extract a behavior log of the malware. Our malware-analysis technology is used for this task.
(3) Generation of IOC candidates with multiple abstractions from malware-behavior logs: Generate regular expressions with various abstractions that can be candidates of an IOC from past malware-analysis knowledge.
(4) Calculation of optimal IOC set based on detection accuracy and ease of interpretation: For each of the above-generated IOC candidates,

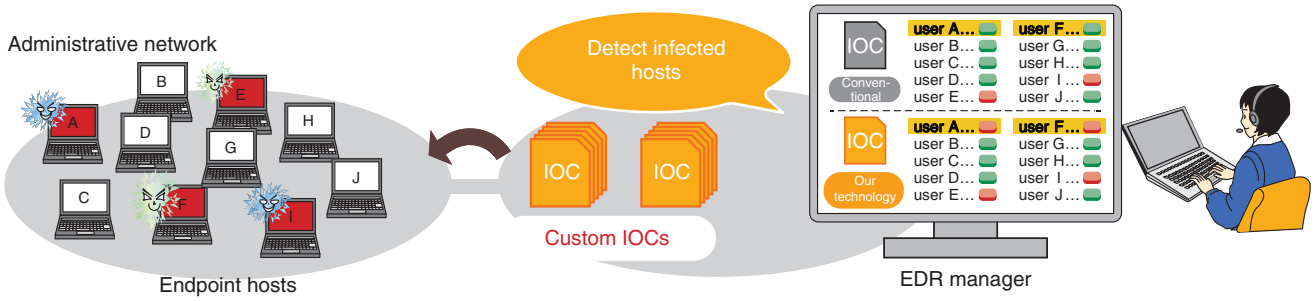


Fig. 2. Use of custom IOCs.

consider detection accuracy and ease of interpretation on the basis on the behavior logs of legitimate software and malware and calculate the optimal IOC for each malware family.

By adding the generated IOC to EDR products on the market, it will be possible to detect malware-infected terminals that have been conventionally difficult to detect (**Fig. 2**). Currently, we collect and select about 10,000 samples per week, and we have started distributing the IOCs generated from the results of analysis of that collection of malware to the NTT Group. In the future, we will continue to respond to malware other than those with executable file formats (such as script format).

4. Current state of public-server defense

Whenever a new vulnerability is discovered on a server or application, cyberattacks target that vulnerability. Cyberattacks that exploit vulnerabilities of servers and applications have exceeded 10 million per day worldwide. To detect and block these attacks, it is becoming common to deploy security devices such as an intrusion prevention system (IPS)[*1] or web-application firewall (WAF)[*2]. Ideally, these security devices should correctly detect and block all attacks. Realistically, however, it is difficult to do this. The reason is that quality of service will degrade due to false blocking. If the security device is not tuned sufficiently by the operator, normal communication may be detected and blocked. Due to this risk, it is difficult to block all attacks without tuning by the operator. Therefore, only detection, such as intrusion detection system (IDS)[*1], is conducted in most cases.

5. The need for more-efficient security operations

A computer-security-incident response team (CSIRT) or security operation center (SOC) analyst responds to cyberattacks occurring in a company or organization. CSIRTs and SOC analysts analyze data daily for security breaches on the basis of alerts sent from security devices.

In particular, WAFs and IDSs that detect server attacks report thousands or tens of thousands of alerts every day, and violations are analyzed on the basis of the knowledge and experience of the analysts. As a result, if the alerts are not prioritized to the more-important ones, it will not be possible to handle all alerts generated in a limited time span.

This prioritization is something that can only be done by a few analysts with knowledge and experience, and now that attacks are growing in scale, it is not practical to analyze all attacks entirely manually because not everyone can do that analysis. Moreover, the attacker can adopt a tactic that requires only a single instantaneous attack of choice to achieve the purpose of the attack while launching many meaningless attacks. Such a tactic paralyzes corporate security monitoring, preventing CSIRT and SOC analysts from noticing the real attack in a timely manner.

6. Alert-triage technology

At NTT Secure Platform Laboratories, we have

*1 IPS/IDS: A system that protects applications from attacks that exploit vulnerabilities; IDS refers to a usage mode that performs only detection, while IPS refers to a usage mode that blocks detected attacks.

*2 WAF: A system that protects applications from attacks in a similar manner to IPS/IDS. It has a detection capability specialized for web applications.

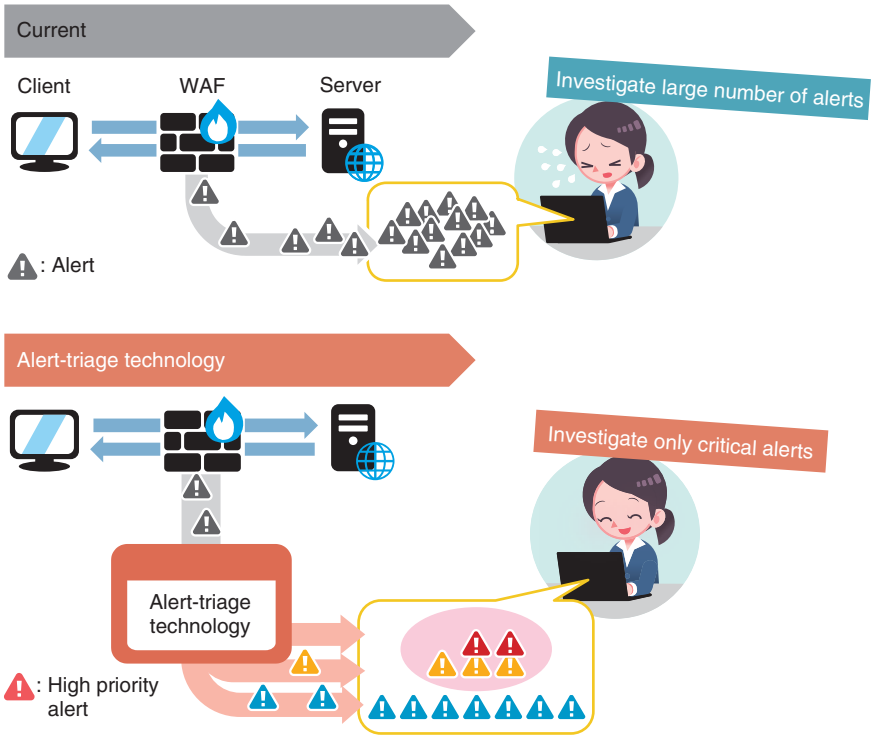


Fig. 3. Alert-triage technology.

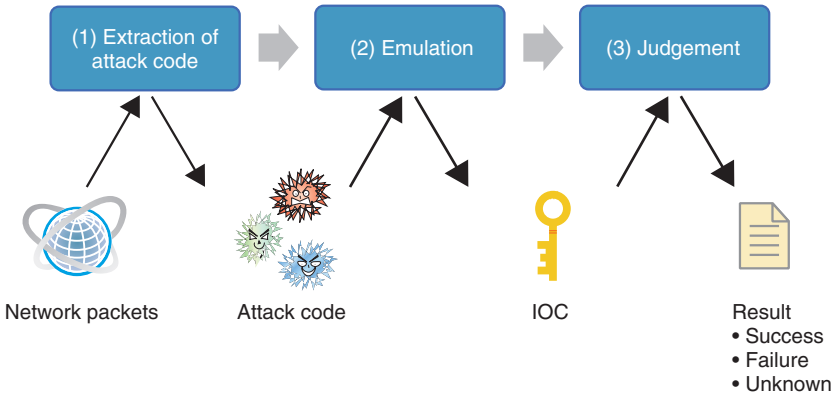


Fig. 4. Basic functions of alert-triage technology.

developed an alert-triage technology [2, 3] that automatically determines the success or failure of an attack on a server (from a network communication) from the trace of the attack and determines whether the alert associated with that attack should be given priority. This is the world's first technology for performing triage (prioritization) by focusing on the success or failure of an attack. This technology enables a server administrator or SOC analyst to concentrate on attacks that require immediate response (**Fig. 3**). The basic functions of this technology are listed as follows (**Fig. 4**).

(1) Extracting the code or command that the attacker wants to execute on the server if the attack is successful
(2) Executing the extracted attack code or command

Time	Alert	Source	Destination	Attack result*	Response priority*
Dec. 2, 2019 21:38:12	Remote code execution attack detected	x.x.x.x	y.y.y.y	Failure	Low
Dec. 2, 2019 21:43:03	SQL injection attack detected	x.x.x.x	y.y.y.y	Success	High
Dec. 2, 2019 21:43:15	Cross-site scripting attack detected	x.x.x.x	y.y.y.y	Undeterminable	Medium

* Information added by our technologies

Fig. 5. Examples of the alert triage.

on an emulator simulating various servers and extracting the trace of the attack (i.e., IOC)

(3) Checking whether the IOC extracted from the emulator was occurring in the actual communication and judging that the attack was successful if it occurred or failed if it did not occur.

When an alert occurs, it is possible with this technology to add information to the alert that (i) the attack was successful, (ii) the attack failed, or (iii) the success or failure of the attack is undeterminable (**Fig. 5**). If the alert has information that indicates the attack was successful, the response priority is high, and the alert should be checked first, even if other alerts are checked later. Conversely, if information is added to the alert that the attack failed, the priority of the response is low, and the alert could be checked later. This technology makes it clear at a glance which alerts should be prioritized, and we believe that if a large event or forum suffers an increasing number of alerts or if an attacker carries out a campaign, the effect of alert-triage technology will be more pronounced.

According to our evaluation using a real network environment, about 52% of alerts were correctly judged as failed attacks, and the priority of response to those alerts was reduced. Moreover, it was possible to raise the priority of related critical alerts to just 0.1% of successful attacks lost in a large number of alerts. An example of another favorable result is that it was also possible to recognize an attack as successful at the reconnaissance stage (at which the damage was still minimal) and, by notifying the operator, to take measures before the attack damage spread.

7. Future developments

Under the supposition that it is practically difficult to prevent cyberattacks in advance, a technology that determines the success or failure of a malware infection at an endpoint terminal or an attack on a public server—by focusing on the traces left during the attack—was introduced in this article. For future work, we will advance our research on technology for automating the response after detection of an attack to counter cyberattacks—which are expected to become increasingly sophisticated and numerous.

References

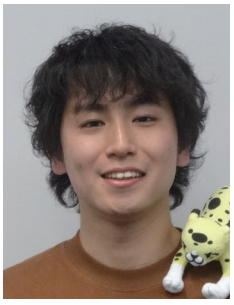
[1] Y. Kurogome, Y. Otsuki, Y. Kawakoya, M. Iwamura, S. Hayashi, T. Mori, and K. Sen, "EIGER: Automated IOC Generation for Accurate and Interpretable Endpoint Malware Detection," The Annual Computer Security Applications Conference (ACSAC) 2019, San Juan, USA, Dec. 2019.

[2] Y. Kanemoto, K. Aoki, J. Miyoshi, H. Shimada, and H. Takakura, "Detecting Successful Attacks against Web Application based-on Attack Code Emulation," IPSJ Journal, No. 60, Vol. 3, pp. 945–955, 2019 (in Japanese).

[3] Y. Kanemoto, K. Aoki, M. Iwamura, J. Miyoshi, D. Kotani, H. Takakura, and Y. Okabe, "Detecting Successful Attacks from IDS Alerts Based on Emulation of Remote Shellcodes," Proc. of the IEEE Computer Society's signature conference on Computers, Software and Applications (COMPSAC) 2019, Vol. 2, pp. 471–476, Milwaukee, USA, July 2019.

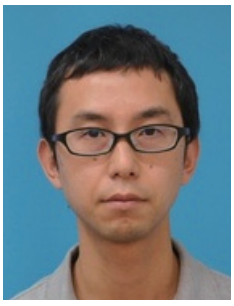
Makoto Iwamura
Distinguished Researcher, NTT Secure Platform Laboratories.
He received a B.E., M.E., and D.Eng. in science and engineering from Waseda University, Tokyo, in 2000, 2002, and 2012. He joined NTT in 2002. He is currently with NTT Secure Platform Laboratories, where he is engaged in the Cyber Security Project. His research interests include reverse engineering, vulnerability discovery, and malware analysis.

Yo Kanemoto
Researcher, NTT Secure Platform Laboratories.
He received a B.E. and M.E. in information science from Nagoya University in 2011 and 2013. Since joining NTT in 2013, he has been researching and developing network security technologies.



Yuma Kurogome
Researcher, NTT Secure Platform Laboratories.
His research interest lies in both defensive and offensive perspectives on malware, that is, malware detection/analysis and evasion. His major research results were presented at industrial venues such as CODEBLUE'15 and Black Hat Europe'19 Arsenal, and an academic paper for ACSAC'19.

Kazufumi Aoki
Senior Research Engineer, NTT Secure Platform Laboratories.
He received a B.E. and M.E. in information science from Tohoku University, Miyagi, in 2004 and 2006. Since joining NTT in 2006, he has been researching and developing cybersecurity technologies. His research interests include reverse engineering and malware analysis.



Yuhei Kawakoya
Distinguished Researcher, NTT Secure Platform Laboratories.
He received a B.E., M.E., and D.Eng. in science and engineering from Waseda University, Tokyo, in 2003, 2005, and 2019. Since joining NTT in 2005, he has been involved in computer system security research. From 2013 to 2016, he was engaged in R&D at NTT Innovation Institute, Inc., USA, as a software engineer. His research interests include reverse engineering, program obfuscation, malware analysis, and forensics.

Shingo Orihara
Senior Research Engineer, NTT Secure Platform Laboratories.
He received a B.E. and M.E. in information science from Tohoku University, Miyagi, in 2001 and 2003. Since joining NTT in 2003, he has been researching and developing network security technologies. His research interests include security log analysis and cloud computing.

Jun Miyoshi
Senior Research Engineer, Supervisor, Cyber Security Project, NTT Secure Platform Laboratories.
He received a B.E. and M.E. in system science from Kyoto University in 1993 and 1995. Since joining NTT in 1995, he has been researching and developing network security technologies. From 2011 to 2016, he was engaged in R&D strategy management of NTT Secure Platform Laboratories. He is currently a research group leader of Cyber Security Project in the Laboratories. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).