Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.034287
Article

An Effective Threat Detection Framework for Advanced Persistent Cyberattacks

So-Eun Jeon[1], Sun-Jin Lee[1], Eun-Young Lee[1], Yeon-Ji Lee[2], Jung-Hwa Ryu[2], Jung-Hyun Moon[2], Sun-Min Yi[2] and Il-Gu Lee[1,2,*]

[1]Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul, 02844, Korea
[2]Department of Convergence Security Engineering, Sungshin Women's University, Seoul, 02844, Korea
*Corresponding Author: Il-Gu Lee. Email: iglee@sungshin.ac.kr
Received: 22 June 2022; Accepted: 08 February 2023

Abstract: Recently, with the normalization of non-face-to-face online environments in response to the COVID-19 pandemic, the possibility of cyberattacks through endpoints has increased. Numerous endpoint devices are managed meticulously to prevent cyberattacks and ensure timely responses to potential security threats. In particular, because telecommuting, telemedicine, and tele-education are implemented in uncontrolled environments, attackers typically target vulnerable endpoints to acquire administrator rights or steal authentication information, and reports of endpoint attacks have been increasing considerably. Advanced persistent threats (APTs) using various novel variant malicious codes are a form of a sophisticated attack. However, conventional commercial antivirus and anti-malware systems that use signature-based attack detection methods cannot satisfactorily respond to such attacks. In this paper, we propose a method that expands the detection coverage in APT attack environments. In this model, an open-source threat detector and log collector are used synergistically to improve threat detection performance. Extending the scope of attack log collection through interworking between highly accessible open-source tools can efficiently increase the detection coverage of tactics and techniques used to deal with APT attacks, as defined by MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK). We implemented an attack environment using an APT attack scenario emulator called Carbanak and analyzed the detection coverage of Google Rapid Response (GRR), an open-source threat detection tool, and Graylog, an open-source log collector. The proposed method expanded the detection coverage against MITRE ATT&CK by approximately 11% compared with that conventional methods.

Keywords: Advanced persistent threat; cybersecurity; endpoint security; MITRE ATT&CK; open-source threat detector; threat log collector

1 Introduction

Cyberattack technology has kept pace with advancements in cutting-edge information technology (IT) [1]. According to "The AV-TEST Security Report 2019/2020" published by the AV-TEST Institute, a German IT security research institute, 114,312,703 malicious codes were created in 2019 and typically distributed through e-mails globally. More than 43 million samples of newly registered malicious codes were created in the AV-TEST antivirus and security software suite in the first quarter of 2020 alone. Thus, instances of cyberattacks have increased considerably [2]. Advanced persistent threats (APTs) mostly targeting large organizations and governments have evolved to be fast and sophisticated, rendering their detection difficult [3].

Although cyberattack techniques and security technologies for countermeasures are evolving simultaneously, most security technologies are developed ex post facto. Therefore, a single unidentified vulnerability can expose systems to intruders, causing attackers to search for vulnerabilities. In most cases, endpoint devices, such as desktop computers, laptops, tablets, and smartphones [4], are the access points for hackers because of their unprotected open environments and connection to networks.

Recently, the proportion of non-face-to-face online environments and *untact* (doing things without direct contact with others) consumption, such as telecommuting, telemedicine, and tele-education, has increased owing to COVID-19. Consequently, endpoints have become increasingly vulnerable to attacks. For example, attack opportunities have increased because widespread telecommuting has boosted the open use of internal network resources across uncontrolled external environments [5]. Among cyberattacks, APTs are difficult to detect. An APT attack refers to covert access to a computing infrastructure within an organization, by which attackers use sophisticated and diverse security threats to induce potentially destructive results. The attacker performs a procedure-based attack to maintain persistency.

With the continuous evolution of intelligent attacks, such as ransomware and fileless malware attacks, conventional antivirus and anti-malware solutions cannot provide satisfactory attack detection. Hence, to improve the detection of sophisticated cyberattacks, attack detection methods using various threat detectors, such as event monitoring, log collection, and remote forensic tools, have been developed. However, most commercial threat detection systems are developed on closed platforms, rendering system adaption to each endpoint environment difficult. Consequently, it is expensive and time-consuming to develop and implement endpoint security systems [6]. Furthermore, detecting APT attacks with a single open-source threat detector is difficult because conventional systems can detect only fragmentary attacks based on limited log data.

In this study, we considered methods to effectively detect APT attacks through open-source threat detectors by using Google Rapid Response (GRR), a major remote live forensics tool, and Graylog, an interworking data collector to collect various threat logs. GRR is a highly scalable open-source remote live forensics tool that supports fast attack detection and large-scale remote analysis. However, there is a disadvantage in that the data types that can be collected are limited because its output plug-in is limited. Therefore, we intend to expand detection performance by interworking GRR with Graylog, an open-source threat log collection and analysis tool that enables various log collections based on Syslog. Additionally, the detection coverage of the Carbanak attack, an APT strike that actually did occur, was examined in this study by dividing the total number of cases that may occur in the attack pattern by the number of conditions. In other words, we considered the number of instances of various attack patterns that can occur in standardized APT assault patterns to determine the detection coverage. The expanded detection coverage of the proposed method was derived by mapping procedure-based attack

actions of APT attacks to cyberattack tactics and techniques defined in MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK).

The contributions of this study are as follows:

- First, to improve the data collection capabilities of conventional open-source threat detectors, the detection and response coverage for APT attacks were expanded by interworking with open-source threat detectors and log collectors. Although various commercial threat detection methods have been developed, recent studies have typically used closed-platform-based tools. By contrast, in the proposed method, easily accessible open-source threat detectors are used.
- Second, in this study, we evaluated the attack coverage based on the MITRE ATT&CK framework, in which the procedure-based attack actions of the cyber kill chain are defined. This method can be used as an improved attack detection and response method for APT attacks occurring in the future.
- Third, a feasible detection coverage expansion method is proposed for attack actions to verify its efficacy against actual Carbanak APT attacks in an emulation environment [7].

The remainder of this paper is organized as follows. Section 2 reviews previous studies on APT attack detection and response techniques. Section 3 details the proposed method for expanding detection coverage and presents an analysis of the response coverage of APT attacks by using open-source threat detectors. Section 4 outlines the experiments conducted to verify the effectiveness of the proposed method against APT attacks and analyzes the results obtained. Finally, Section 5 presents the conclusion.

2 Related Work

In this section, studies related to conventional APT attack detection, response techniques, and threat detection systems are detailed. Table 1 lists the various studies along with their corresponding methods and limitations.

Table 1: Previous studies on advanced persistent threat (APT) attack detection and response techniques

Research study	Reference	Method	Limitations
APT attack detection & response	Berady et al. [8]	APT attack's lifecycle is defined as a state machine model.	The initial intrusion process cannot be represented in the state machine model because the attacker is assumed to have built the initial basis. The criteria for the success of the attack are not defined.
	Ma et al. [9]	APT attacks are based on suspicious domain names using a domain graph.	The attacker's malicious domain name and IP address are assumed not to reduce the effectiveness of the attack.
		The relevance between malicious domain names is investigated based on the relationship between domain names and IP addresses.	When the attacker deliberately uses a different IP address, providing a response is difficult, and the detection scalability decreases. The assumed circumstances are far from real circumstances.
	Stojanovic et al. [10]	The necessity for automated detection methods for APT attacks is explained.	Specific detection methods based on dataset creation are not sufficiently explained.

(Continued)

Table 1: Continued

Research study	Reference	Method	Limitations
		An overall lifecycle and attack model for APT attacks is analyzed, and APT attack dataset creation models are reviewed for attack detection.	
	Joloudari et al. [11]	APT attack detection and classification using AI-based classification models are investigated.	There is a lack of comparative analysis with various learning algorithms. Precise APT attack scenarios are not considered.
		The Network Security Laboratory Knowledge Discovery and Data Mining (NSL-KDD) dataset is used.	Training data quality has a significant impact on detection accuracy.
	Mohamed et al. [12]	An anomaly detection model based on strange behavior inspection (SBI) is proposed to detect APT attacks. The detection method is based on a credential dumping method using the vulnerability in the lifecycle of APT attacks. A novel model based on the MITRE ATT&CK metrics is proposed to detect APT attacks.	Detection is difficult in exceptional circumstances during the initial intrusion process.
Detect by utilizing threat detectors	Karantzas et al. [13]	The effectiveness of commercial endpoint detection and response (EDR) products is evaluated for APT attack detection and response. According to the results, current commercial EDR solutions exhibit insufficient performance in terms of responding to APT attacks.	The evaluation of commercial tools is focused on closed platforms only.
	Hassan et al. [14]	The advantage of causal relationship analysis (data provenance) is applied to EDR to overcome the limitations of EDR tools. A novel log reduction scheme that can reduce the storage overhead of system logs while maintaining the causal relationships between time points of threats is proposed.	Experiments are based on hypothetical scenarios simulated under the assumption that no attack occurs between procedures of an APT attack. Record of arrest and prosecution sheet's (Rapsheet) log reduction algorithm assumes that all threat alerts are reduced by the default EDR tool. The real-world environment is not considered.
	Kieseberg et al. [15]	A comparison/analysis of features and functions of open-source remote forensic tools is performed.	The number of distributed clients is limited in the experimental environment, and only the detection of simple malware is analyzed. It is impossible to determine whether the same detection rate can be achieved when APT attacks with sophisticated procedures occur (fragmentary analysis).
	Matsuda et al. [16]	A real-time threat detection method that identifies malicious dynamic link library (DLL) files using Sysmon is proposed. A practical detection method that uses Elastic Stack as Security Information and Event Management (SIEM) is proposed.	A detection method for the case where the characteristics of the malicious DLL are bypassed is not presented. The solution cannot be applied to other operating systems except Windows.

(Continued)

Table 1: Continued

Research study	Reference	Method	Limitations
	Rasheed et al. [17]	GRR, an open-source EDR tool, is used to evaluate the attack detection performance for remote code execution and exploit attacks.	Results are derived through fragmentary attack experiments. A realistic attack environment is not considered.
	Park et al. [18]	An open-source EDR system combining GRR and osquery is evaluated for threat detection, and incident detection experiments are conducted. APT coverage for the proposed EDR system is analyzed using MITRE's adversarial tactics, techniques, and common knowledge model.	The attack environment used is not realistic.

2.1 APT Attack Detection and Response

Studies on APT attack detection have typically focused predominantly on signature-, graph-, and machine learning-based detection methods. Berady et al. [8] proposed a Nuke model in which the APT attack lifecycle was defined as a state machine. This model consists of exploration, exploitation, and decision-making phases. Unlike previous studies, where modeling was performed based on conventional attack strategies, they studied efficient modeling of APT attacks with complex attack procedures by using a state machine to depict repetitive, idle, and withdrawal actions. However, their model had the following limitations. The initial intrusion process of an APT attack could not be represented because the attacker could perform an attack by building an initial intrusion basis. Furthermore, realistic environments were not considered because the criteria for the success of the attack were not defined.

Ma et al. [9] used a domain graph to detect APT attacks based on suspicious domain names and proposed a model investigating the relevance between malicious domain names based on the relationships between domain names and internet-protocol (IP) addresses. They proposed an APT attack detection system that can be applied to sensor networks using a detection method that ensures accuracy and scalability while reducing the computational complexity of the detection system by using a domain graph. However, when the attacker accesses using a different IP address, the method cannot react, and the detection scalability decreases considerably.

Stojanovic et al. [10] explained the necessity for automated detection methods for APT attacks and stated that, although various studies have been conducted on detecting initial intrusion for APT attacks, only a few have focused on detection methods considering lifecycles. Furthermore, they defined the overall lifecycle and attack model of APT attacks and reviewed dataset creation models. However, specific detection methods based on dataset creation were not sufficiently explained in their study.

Joloudari et al. [11] detected and classified APT attacks using three artificial intelligence (AI)-based classification models: Bayesian network, C5.0 decision tree, and deep learning. They achieved improved detection accuracy by using a deep learning model to train and analyze APT attack patterns. However, they conducted experiments using the dataset of a network intrusion detection system and did not consider realistic APT attack scenarios. Furthermore, guaranteeing the effectiveness and accuracy of machine learning-based APT attack detection in real situations is difficult because the detection accuracy varies considerably depending on the quality of the training data.

Mohamed et al. [12] proposed an anomaly detection model based on strange behavior inspection (SBI) to detect APT attacks. Furthermore, they proposed a detection method based on credential dumping using the characteristics of an APT attack that exclude the signature. They also developed a model based on MITRE ATT&CK metrics. However, their detection method focused on only the initial intrusion stage, and detection is difficult when an exceptional circumstance occurs during the initial intrusion process.

2.2 Detection of Cyberattacks by Using Threat Detectors

Karantzas et al. [13] used various attack scenarios to evaluate the efficacy of commercial EDR products for APT attack detection and response. They demonstrated that most EDR tools cannot satisfactorily respond to APT attacks and detailed the necessity for improvement. Focusing on the initial penetration method using attached files in e-mails in typical APT attacks, they conducted experiments by dividing attack vectors by file extension: Control panel (CPL), a control panel setup file extension; Hypertext markup language (HTML) application (HTA), an HTML document execution file extension; Executable (EXE), an application program execution file extension; and DLL, a dynamic link library file extension. The results revealed the performance limitations of conventional commercial EDR tools, which do not effectively respond to APT attacks. They also stated that the evaluation of security logs collected by EDR tools is critical for effectively responding to APT attacks.

Haasan et al. [14] proposed a method that can reduce the storage overhead of system logs and simultaneously maintain causal relationships between threat events to solve the problems of high misdetection rates of conventional EDR tools and loss of logs during long-term attacks. Furthermore, they achieved improved threat detection accuracy by integrating the Symantec EDR tool with Rapsheet, which is a prototype threat detection system. However, they performed simulations and validation under the assumption that no attack occurs between the procedures of an APT attack, which is unlike real-world APT attack scenarios. Furthermore, the real-world environment was not sufficiently considered in the log reduction algorithm, wherein all threat alerts were assumed to be reduced by the default EDR tool.

Kieseberg et al. [15] compared the features and functions of Google's GRR, Facebook's osquery, and Mozilla's InvestiGator, representative remote live forensic solutions of Google, Facebook, and Mozilla, respectively, and analyzed whether these programs accurately detected malicious programs. These companies routinely check thousands of computers. The results of their study revealed that GRR is an optimized EDR tool for analyzing changes caused by attacks. Furthermore, GRR has excellent performance in terms of integrity checks, which are used to examine file contents suspected of being malicious and to detect whether files have been changed in the entire operating system (OS) structure. However, the number of distributed clients was small in their experimental environment, and they analyzed the detection of simple malware only, which renders response to sophisticated APT attacks difficult.

Matsuda et al. [16] proposed a real-time threat detection method using system monitor (sysmon), a free logging tool, to detect sophisticated targeted attacks. They proposed the detection method using DLL because existing threat detectors cannot accurately detect the type of attack that bypasses detection by changing or rebuilding the file name of a malicious tool. Their detection method was based on the feature that any DLL loaded by a malicious tool can be identified according to the version of Windows and the malicious tool. However, a detection method for the case of an attack type when the characteristics of the malicious DLL are bypassed was not presented, and the solution cannot be applied to other operating systems, except for Windows.

Rasheed et al. [17] used GRR to evaluate the attack detection performance for remote code execution and exploit attacks. In their study, detection experiments were conducted for two attack scenarios, and the results revealed that GRR exhibits excellent performance in collecting and analyzing data from multiple clients. However, its functions were inadequate in terms of analyzing collected data, and an external tool with scripts was utilized simultaneously to use the collected data.

Park et al. [18] integrated open-source security frameworks combining GRR and osquery to expand detection coverage. The APT coverage for the proposed EDR system was analyzed using MITRE's adversarial tactics, techniques, and common knowledge model. They subsequently proposed an attack coverage expansion method that can be used in the real world through open-source threat detectors. However, realistic APT attack environments were not considered.

Additionally, machine learning-based detection methods have been studied recently [19]. A study by Chen et al. [19] discussed how to detect APT attacks in IoT environments. This prior study raises the problem that it is challenging to apply ML-based detection methods due to the lack of datasets for APT attacks. Although there are contributions to reviewing 14 different AI-based approaches to detect APT attacks in IoT environments, the dataset analyzed in this previous study has limitations in that it deals with datasets that are difficult to see as APT attack datasets. Even if the dataset is not utilized, discussing approaches that use commercial tools to collect attack logs and perform ML-based detection is necessary. Also, recently, studies have been conducted to detect threats using commercial tools [20]. This preceding study analyzes various open-source tools for attack detection and demonstrates that each tool can quickly identify and respond to APT attacks. Although this previous study contributes to the use of open-source tools by demonstrating the attack detection method of each open-source tool, it is insufficient to prove whether it can respond to precise attacks in that it utilizes a simple and general ransomware attack model.

Furthermore, although numerous studies have been conducted on APT attacks, most could not simulate environments of sophisticated APT attacks occurring in the real world. A systematic attack detection framework is yet to be proposed. Furthermore, researchers have analyzed commercial threat detectors developed on closed platforms, which limits their application to many user environments. Additionally, most studies on threat detectors evaluated threats based on the detection of fragmentary attacks, such as malware detection, which differs considerably from real-world APT attack environments.

In this study, an APT attack environment was constructed using the Carbanak model, and the APT attack detection and response coverage of GRR, an open-source threat detector, was analyzed. The coverage was evaluated based on the tactics and techniques of MITRE ATT&CK, which is a proven framework. To analyze the limitations of GRR and make improvements, a method that provides expanded coverage by interworking with Graylog, a conventional open-source threat log collector, was proposed.

3 Interworking Open-Source Threat Detector and Log Collector

In this section, the functions and operations of conventional open-source threat detectors were analyzed, and a method for expanding the coverage of APT attack detection by interworking open-source threat detection and data collection tools was proposed.

3.1 Open-Source Threat Detector Analysis

In this section, the capability and performance of various commercial open-source threat detection systems were analyzed. The main open-source threat detectors considered were open-source host intrusion detection system (HIDS) security (OSSEC), TheHive, and Whids [21], which were analyzed by using Kieseberg et al.'s comparison metrics [15].

In most open-source threat detectors, the status of connected endpoint devices, including host statistics, process listings, and connected users, is monitored; however, numerous differences exist in the logs that can be detected between the tools. Furthermore, commercial open-source tools cannot collect various logs because of their open-source nature. For example, TheHive does not support Windows registry collection, and neither OSSEC nor TheHive have memory inspection capabilities. In particular, supported operating systems differ for each open-source threat detector, and their use is limited by the environment of inspected devices. Furthermore, threat detectors, such as TheHive, should have an environment that can interwork with Cortex, and using Whids in the Linux or Mac environment is not possible because this threat detector is focused on inspecting Windows systems. By contrast, GRR, which is a remote live forensics tool, can inspect various operating systems, and analysts can easily and efficiently use it to analyze large networks. Furthermore, automated analysis can be performed over a broad range by scheduling tasks in advance.

Therefore, in this study, we performed our analysis on GRR, which can be operated in various operating systems and allow for broad monitoring through unique artifact collection. GRR is used as a threat detector for APT attack detection. It is a leading open-source threat detector that provides fast and scalable capabilities for fast attack classification and large-scale remote analysis. By contrast, collecting various logs to achieve a high detection effect is crucial; however, it exhibits a limited log collection capability because of the nature of the open-source tool, which renders detection and response to sophisticated APT attacks difficult. Therefore, a detection coverage expansion method was proposed to create an interworking environment with Graylog, a commercial open-source threat log collector.

3.2 Open-Source Threat Detection and Data Collection Analysis

3.2.1 GRR Analysis

GRR is a remote live forensics tool developed by Google, which supports fast and scalable inspection for performing remote analysis after rapidly classifying attacks based on an intrusion incident response framework focused on remote live forensics [22]. GRR operates in a client–server architecture.

GRR consists of clients, inspected devices; a GRR front-end server that sends and receives messages to and from clients; workers, which divide and process flows scheduled on the server; the advanced forensics file format (AFF4) subsystem; and a data store for accumulating data. Clients exchange messages with the server through the hypertext transfer protocol (HTTP), and the server communicates with the AFF4 subsystem to save data in the data store. GRR collects various forensics artifacts from multiple operating systems and uses the flow function to collect logs, including those of the registry, remote file, input/output, usage, central processing unit (CPU), and memory information. Flow is used to collect data at the artifact level for attack detection, and the hunt function is used for fast analysis when inspecting large-scale agents.

3.2.2 Graylog Analysis

Graylog operates on humongous database (MongoDB) and Elasticsearch and is a highly available open-source log collection and management platform for Syslog [23], which is a data collection tool optimized for collection. Syslog facilitates convenient and simultaneous checking of logs of all connected devices through the centralization of logs. Furthermore, Windows event logs are fetched by using Nxlog in the Windows environment. Because Graylog classifies risk levels for each log, attack situations can be determined based on the collected logs. In the case of GRR, logs are collected and analyzed by focusing on the artifacts and memory analysis of the file system. Therefore, to supplement this, the detection coverage of APT attacks was analyzed by interworking with Graylog, a Syslog data collection tool that comprehensively collects logs.

3.3 Interworking Open-Source Remote Live Forensics and Threat Log Collection for APT Attack Detection Coverage Expansion

3.3.1 Structure of the Proposed Model

In the proposed method, the detection coverage of APT attacks is expanded by collecting various logs through the interworking of an open-source threat detector and threat log collector. GRR has excellent capability of collecting various unique artifacts and provides the hunt function for large-scale inspection. However, the method has limited scope and collectible data types because its output plugins are limited. Furthermore, the method does not focus on log collection. To solve this problem, a method of expanding coverage was provided by interworking with Graylog, a syslog-based open-source threat log collection and analysis tool, as shown in Fig. 1.

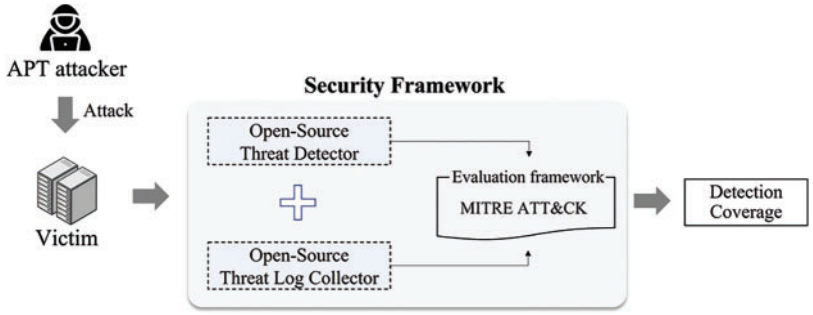

Figure 1: Structure of the proposed scheme

Brown et al. [24] analyzed a method of interlinking the output plugin of the GRR and Graylog extended log format to improve the limitations of GRR. However, they did not clearly describe the parts specifically improved through the interworking of the threat detector and threat log collector. In this study, we evaluated the detection coverage of the open-source threat detector based on a sophisticated APT attack scenario and analyzed the coverage by assuming an interworking environment with Graylog.

Based on the tactics and techniques of attackers (MITRE ATT&CK framework), a security framework based on information on cyberattack tactics and techniques was defined. Furthermore, the patterns of APT attacks in terms of a state machine were defined to comparatively analyze the detection coverage for each of various attack patterns.

3.3.2 Carbanak APT Attack

An attack environment was implemented using an APT attack scenario emulator of the Carbanak Group, a firm that has been active since 2013. The Carbanak attack scenario consists of an attack system (Attack Platform) and four victim systems (i.e., Human Resources (HR) manager, BankFileServer, Domain Controller, and Chief Financial Officer (CFO) User). Here, HRmanager refers to a victim system in which the APT attack begins first by executing a malicious file and corresponds with bank employees in a common financial service environment. BankFileServer refers to the target in which the attacker sends a lateral movement tool to expand the attacking range along with bank file servers. Domain Controller refers to a domain administrator who manages the domain of employees inside the organization, and CFO User refers to the final target system of the attack and corresponds to the person responsible for finance. Fig. 2 shows the specific attack procedures of the Carbanak attack scenario.

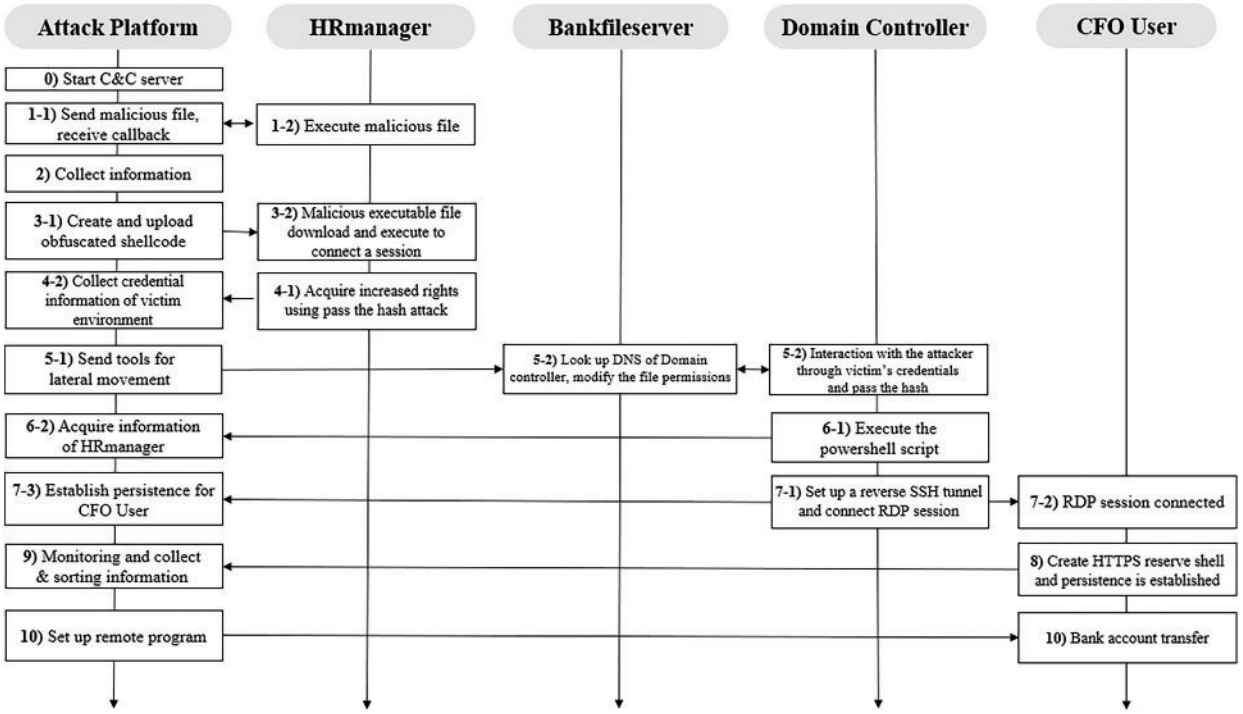

Figure 2: Attack flow of Carbanak

The Carbanak scenario is an 11-step APT attack procedure from Step 0 to Step 10, where the attacker achieves the final goal by transferring funds to a malicious account. The attacker operates the command and control (C2) server to launch an attack (Step 0) and acquires information, such as the host name, username, and CPU architecture, of the victim system by initially penetrating the HRmanager's computer (i.e., by uploading a malicious file to an internal employee's computer) (Step 1, Step 2). Next, the attacker creates an obfuscated shellcode to connect a session (Step 3). Subsequently, after executing the Powerview module on the HRmanager's computer to elevate the attacker's access rights, the attacker sends a User Account Control (UAC) bypass script and a malicious executable file to the bank employee's computer (HRmanager). When executed, the attacker can collect the credential information of the victim (Step 4). The attacker then uploads tools to infiltrate CFO User (the target system) to expand the access domain. At this stage, the attacker sends a lateral movement tool to

HRmanager and BankFileServer, which is used to connect to the secure shell (SSH) to look up the domain name system (DNS) of Domain Controller. Next, the file permissions are changed so that the transmitted file can be executed. The file is then executed by moving the shell of Domain Controller to enable interaction with the attacker (Step 5). By executing the PowerShell script on Domain Controller, the attacker acquires the information of CFO User, the final target system (Step 6). The attacker establishes a reverse SSH tunnel on CFO User to maintain persistence and connects a remote desktop protocol (RDP) session on Domain Controller by connecting a session to CFO User's RDP (Step 7). When CFO User logs in, a secure HTTP reverse shell is created, providing a foundation for the attacker to gather information about the final target system (Step 8). If CFO User executes the payment transfer system, the attacker can collect and monitor the information of artifacts (Step 9). The attacker finally impersonates the CFO User and directly interacts with the fund transfer system to transfer money to a malicious account, thereby achieving the goal of the attack (Step 10).

3.3.3 Definition of the APT Attack State Machine

The APT attack proceeds as follows: Intelligence gathering is performed to obtain information about the attack target organization; the initial compromise step is used for initial penetration; the command and control (C&C) communication step is performed to maintain persistent communication with the primary infected device; lateral movement step is used to expand the access domain; asset discovery step to obtain additional information about the target for further data exfiltration in the future; and data exfiltration step was performed to finally exfiltrate data [25].

The patterns of APT attacks that may occur were defined by transforming the flow of the Carbanak APT attack as a state machine, as shown in Fig. 3.

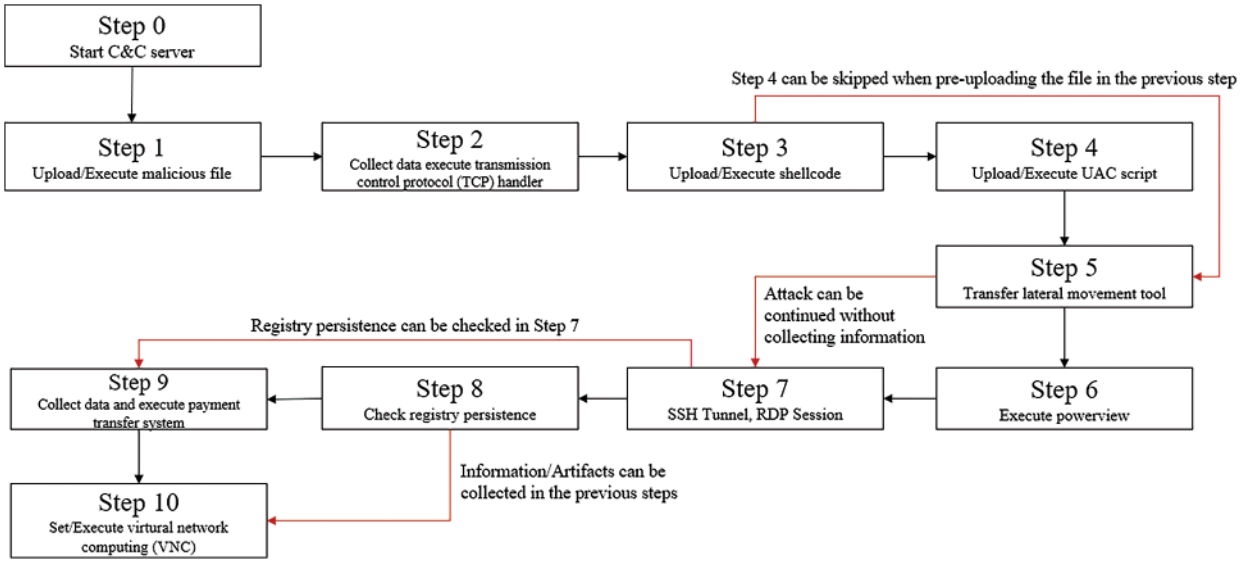

Figure 3: Carbanak attack state machine

Step 4 sends a malicious executable file to the computer of the victim to elevate the rights, and because it can be performed during the previous step of session connection, the attacker can immediately expand the attack from Steps 3 to 5. Furthermore, in Step 6, the final attack target system's information is obtained from Domain Controller. This step also can be skipped to Step 7, assuming that the information has been acquired from the previous step. In Step 8, the registry persistence is

first executed and then Step 9 is executed, assuming that it was executed during the registry-setting Step 7. Step 9 can be skipped and Step 10 moved to immediately, if the victim's information has been collected and monitored at the previous step.

For this study, we extracted 16 APT attack types based on the state machine and analyzed the detection coverage of the proposed model for each pattern.

4 Experimental Evaluation

4.1 Experimental Environment

In this study, we implemented an experimental environment of APT attacks using a Carbanak emulator [7], which is available on Github. Next, the attack detection coverage based on the interworking of GRR and Graylog were analyzed as the threat log collector and attack detector, respectively. Fig. 4 shows the system environment used in the experiment.

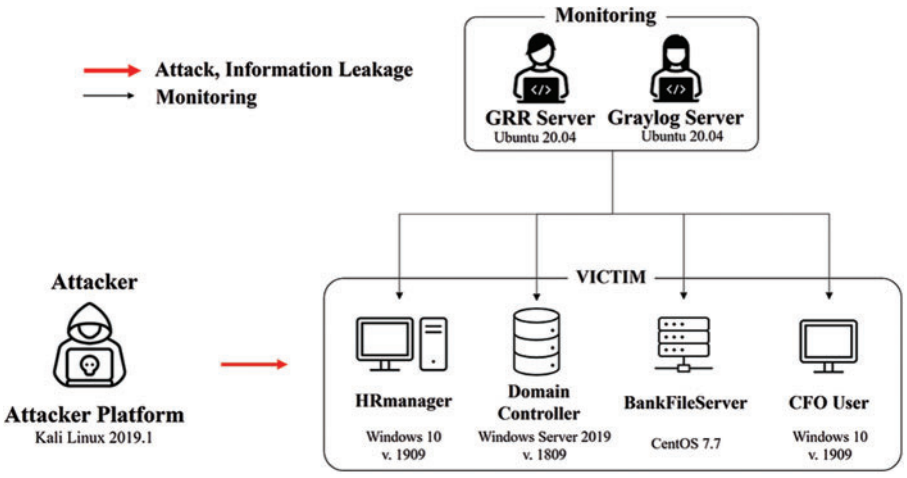

Figure 4: Environment of attack and detection experiment

The attack experiment was conducted on five virtual machines. The attacker system performed the attack in Kali Linux 2019.1, the victim system HRmanager performed the experiment in Windows 10 1909, Domain Controller in Windows Server 2019 1809, BankFileServer in community enterprise operating system (CentOS) 7.7, and CFO User in Windows 10 1909. Furthermore, the GRR and Graylog Server, a monitoring environment for detection, was established in the Ubuntu 20.04 environment to collect attack logs.

The APT attack was applied, and the detection and response model explained in Section 3.3 was used to conduct the attack test in an environment of one attacker platform and four victim systems. Furthermore, the GRR and Graylog used for attack detection were implemented in the environment of four victim systems.

4.1.1 Attack Technique for Each Procedure Based on MITRE ATT&CK

Table 2 maps each step of the attack scenario to the tactics and techniques of MITRE ATT&CK. MITRE ATT&CK is a security framework based on cyberattack tactics and techniques and provides

a framework that can identify the attacker's behavior by analyzing behavior patterns that occur when the attacker interacts with endpoints of the system.

Table 2: Attack tactics in the Carbanak scenario based on MITRE ATT&CK

Step	Summary	Tactics	Techniques
Step 0 (Start C2 Server)	–	–	–
Step 1 (Initial Breach)	User Execution	Initial Access	External Remote Services
		Execution	Command and Scripting Interpreter
		Defense Evasion	Obfuscated Files or Information
			Deobfuscate/Decode Files or Information
		Lateral Movement	Remote Service Session Hijacking
		Command and Control	Application Layer Protocol
Step 2 (Target Assessment)	Local Discovery	Discovery	System Owner/User Discovery
			System Information Discovery
			Process Discovery
	Screen Capture	Execution	Windows Management Instrumentation
			Command and Scripting Interpreter
		Collection	Screen Capture
		Command and Control	Ingress Tool Transfer
		Exfiltration	Exfiltration Over C2 Channel
Step 3 (Deploy Toolkit)	Stage 2 Remote Access Trojan (RAT)	Defense Evasion	Obfuscated Files or Information
			Modify Registry
	Execute 2nd stage RAT	Execution	Command and Scripting Interpreter
		Defense Evasion	Deobfuscate/Decode Files or Information
			Process Injection
		Discovery	Query Registry
		Command and Control	Ingress Tool Transfer
			Non-Standard Port
Step 4 (Escalate Privileges)	Local and Domain Discovery	Discovery	File and Directory Discovery
			Remote System Discovery
			Permission Groups Discovery
	UAC Bypass and Credential Dumping	Execution	Command and Scripting Interpreter
		Privilege Escalation	Abuse Elevation Control Mechanism
		Credential Access	OS Credential Dumping
		Command and Control	Ingress Tool Transfer
Step 5 (Expand Access)	Ingress and Lateral Tool Transfer	Lateral Movement	Lateral Tool Transfer
		Command and Control	Ingress Tool Transfer
	Lateral Movement via SSH	Discovery	Process Discovery
			File and Directory Discovery
		Lateral Movement	Remote Services
	Lateral Movement via PsExec + Pass-the-Hash	Execution	System Services
		Defense Evasion	Use Alternate Authentication Material
		Lateral Movement	Remote Services
Step 6 (Discover Potential Targets)	Remote System Discovery	Discovery	Remote System Discovery
			Account Discovery

(Continued)

Table 2: Continued

Step	Summary	Tactics	Techniques
Step 7 (Setup Persistence)	RDP through Reverse SSH Tunnel	Lateral Movement	Remote Services
		Command and Control	Protocol Tunneling
	Lateral Movement to CFO via RDP	Persistence	Valid Accounts
		Discovery	System Owner/User Discovery
		Lateral Movement	Remote Services
	Registry Persistence	Persistence	Boot or Logon Autostart Execution
Step 8 (Gain Covert Access to Target)	Execute Registry Persistence on CFO	Persistence	Boot or Logon Autostart Execution
		Command and Control	Application Layer Protocol
Step 9 (Profile a Victim User)	User Monitoring	Credential Access	Input Capture
		Discovery	Process Discovery
		Collection	Screen Capture
	Credentials from Web Browsers	Defense Evasion	Indicator Removal on Host
		Credential Access	Credentials from Password Stores
Step 10 (Impersonate Victim)	Install Virtual Network Computing (VNC) Persistence	Privilege Escalation	Create or Modify System Process
		Defense Evasion	Impair Defenses
			Modify Registry
		Command and Control	Ingress Tool Transfer
	Use VNC Persistence	Privilege Escalation	Valid Accounts
		Lateral Movement	Remote Services

The MITRE ATT&CK-based tactics and techniques mapped in Carbanak navigator [26] on Carbanak emulator Github [7] were remapped based on the latest version of MITRE ATT&CK "ATT&CK V9" [27]. Next, we calculated the coverage based on the success/failure of detecting the mapped tactics and techniques.

4.1.2 Coverage Calculation Method

The following equations based on the mapped tactics and techniques presented in Table 2 were applied to calculate the detection coverage of the detection system—GRR and Graylog. Eq. (1) calculates the coverage for tactics, and Eq. (2) calculates the coverage for techniques.

$$\frac{\sum_{i=0}^{T} \frac{Number\ of\ dectected\ tactics\ among\ the\ same\ tactics}{Number\ of\ the\ same\ tactics}}{Number\ of\ tactic\ types\ within\ the\ step} \times 100(T = Number\ of\ tactic\ types\ within\ the\ step) \tag{1}$$

$$\frac{Number\ of\ detected\ techniques\ among\ the\ same\ tactics}{Number\ of\ the\ same\ tactics} \times 100. \tag{2}$$

The formula for calculating the coverage of Tactics, Eq. (1), is used to calculate the number of detected tactics among the same tactics as a percentage of the number of tactics mapped for each step in the Carbanak scenario. Similarly, the formula for calculating coverage, Eq. (2), calculates the number of techniques detected among the same tactics as a percentage of the number of tactic types mapped to each step of the Carbanak scenario.

Table 3 details the tactics and techniques mapped in Step 3 in the Carbanak attack scenario and the logs for describing examples of the coverage calculation methods of Eqs. (1) and (2). Here, "$\surd$" denotes that the attack behavior can be detected through the logs collected at each step, and

"X" denotes detecting the attack is difficult because the logs are not sufficiently collected. Furthermore, "Δ" denotes that only some of the logs were collected in each tactic.

For tactics, calculations were performed based on the number of tactics mapped in each step, and the "same" in each equation refers to the number of certain tactics mapped in the step.

Table 3: Results of the detection experiment in step 3

Tactics	Techniques	Collected log type		Detection
Defense Evasion	Obfuscated Files or Information	Process	List-Processes	√
Defense Evasion	Modify Registry	Registry	Registry Finder	Δ
Command and Control	Ingress Tool Transfer	Virtual File System	(VFS)	√
		Filesystem	File Finder	
Execution	Command and Scripting Interpreter	Processes	Client-Side File Finder	Δ
Discovery	Query Registry	–		X
Defense Evasion	Deobfuscate/Decode Files or Information	–		X
Defense Evasion	Process Injection	–		X
Command and Control	Non-Standard Port	Network	Netstat	√

In Step 3, which connects sessions with the victim system, log collection related to Query registry, Deobfuscate/Decode Files or Information, and Process injection techniques was insufficient when using the GRR and Graylog combination. However, the proposed log severity score-based detection rule makes it possible to respond quickly to APT attacks through log collection related to other techniques.

For example, in the case of Step 3 in Table 3, the number of tactic types in the step, which corresponds to the denominator, is four because four types exist: Defense Evasion, Command and Control, Execution, and Discovery. In the case of the numerator, the number of tactics detected is calculated based on the same tactic. In the case of Step 3, the numerator is "1/4" for the four tactics defined as Defense Evasion because one attack was successfully detected; the numerator is "1" for the two tactics defined as Command Control because two attacks were successfully detected. By expressing this as a ratio, the tactics coverage can be calculated as follows:

$$\textit{Tactics coverage in Step } 3 = \frac{5/4}{4} \times 100 = 31.25\%. \tag{3}$$

Techniques refer to specific attack techniques included in tactics, and the coverage is derived by converting the number of the detected techniques to a percentage of the number of the same tactics. When calculating the coverage of the techniques in the case of Step 3 in Table 3, the denominator, as the number of each tactic type, is "4" for Defense Evasion, "2" for Command and Control, "1" for Execution, and "1" for Discovery. In the case of the numerator, the number of detected techniques is calculated based on the same tactic. In Step 3, the number of detected techniques is "1/4" for the four tactics defined as Defense Evasion because one attack was successfully detected; "2/2" for the two tactics defined as Command and Control because two attacks were successfully detected; and "0/1"

in the case of Discovery and Execution. By expressing these results as ratios, the coverage in terms of techniques can be calculated as follows:

$$Coverage\ for\ the\ techniques\ of\ Defense\ Evasion = \frac{1}{4} \times 100 = 25\% \tag{4}$$

$$Coverage\ for\ the\ techniques\ of\ Discovery = \frac{0}{1} \times 100 = 0\%, \tag{5}$$

$$Coverage\ for\ the\ techniques\ of\ Execution = \frac{0}{1} \times 100 = 0\%, \tag{6}$$

$$Coverage\ for\ the\ techniques\ of\ Command\ and\ Control = \frac{2}{2} \times 100 = 100\%. \tag{7}$$

4.2 Results

When only GRR was used and detection was performed in the environment of interworking with a threat log collector, Graylog, we measured the average coverage of tactics and techniques for the entire steps and each step of APT attacks, from Step 0 to Step 10, based on MITRE ATT&CK and the average coverage for each attack pattern based on the state machine. Graylog was used a data collection tool, registry logs, malicious reverse shell connection logs, and network and file logs were also collected, which extended the overall detection coverage.

Table 4 is a log collected from GRR and Graylog for each Carbanak step.

Table 4: Results of the collected log data of Carbanak

Step	Attacker's action	Collected log data type	
		GRR	Graylog
0	Start C2 server	–	–
1	Execute malicious file	Network log Process log	Network log Process log Account log
2	Collect victim's information	Process log Network log	System log Process log Network log
3	Upload/Execute shellcode	Process log File log Network log	Process log Registry log
4	Upload/Execute UAC script	File log Process log	File log Process log
5	Transfer lateral movement tool	File log Network log Process log	Network log File log Process log Account log
6	Execute Powerview	Process log	Process log
7	SSH tunnel, RDP session	File log Process log	Registry log Network log

(Continued)

Table 4: Continued

Step	Attacker's action	Collected log data type	
		GRR	Graylog
			System log
		Network log	Process log
			Account log
8	Check registry persistence	Network log	Network log
9	Collect data and execute payment transfer system	Network log	Process log
		File log	File log
		Process log	
10	Set/Execute VNC	Process log	Registry log
			Network log
		Network log	Account log

According to Table 4, collecting various APT attack logs through interworking with Graylog, a data collection tool, was possible. Furthermore, various logs are collected even within the same log type, which increases the possibility of detecting APT attacks. For both tools, network and process logs were collected in common, and account, system, and registry logs were additionally collected through Graylog. The coverage of the interworking environment in which the data collected from GRR and Graylog were combined was analyzed.

To derive practical coverage, detection logs were collected in the APT attack environment, and scores based on the severity level of the logs were calculated to determine whether attacks were detected.

4.2.1 Calculation of the Score by the Severity of Log

The standards specified by Microsoft were used for the severity level of logs [28]. The higher the severity level, the higher the possibility of a log being left by the attacker. Level 0 corresponds to a type of log related to general system information, and a log of this level is given a severity score of 0 points. Level 1 is a type of event that can potentially cause a problem, which is not an actual error, but a component or application may experience an error due to additional actions. For example, a log that can detect the changes in the filesystem matched to Level 1. Logs at this level were given a severity score of 1 point. Level 2 is a type of event in a category that has problems, but it does not require immediate action. This result includes logs that can detect abnormal process running. Logs at this level were given a severity score of 2 points. Level 3 is the most severe log level and requires immediate attention from system administrators, typically system and application-level events. Logs that can detect a suspicious port in a network or detect a new connected session are matched in Level 3. As the highest level of severity, the highest severity score of three points was given. In this paper, we propose a method to detect APT attacks by setting a detection ruleset based on the severity of the collected logs. Based on MITER ATT&CK, a security framework composed of various attack tactics and techniques, detection coverage is determined using the severity ruleset of the collected logs that have identified APT attacks. This coverage is then analyzed under the assumption of an accurate detection rate. Therefore, even if a combination of GRR and Graylog fails to collect logs related to specific tactics and techniques, it will be detectable against APT attacks sufficiently early based on the accumulated severity score of the

logs. Additionally, research is being conducted to shorten the APT attack detection times [29,30]. While Park et al. [29] proposed a high-speed detection method to quickly detect and respond to APT attack processes using the open-source tools GRR and Auditbeat, Stoleriu et al. [30] proposed automated detection techniques by integrating with Elasticsearch and VirusTotal to automatically query the hash of malicious files. By integrating several tools, such conventional studies have also been conducted to shorten the time taken to detect attacks.

Based on the severity level scores of logs, the severity score of the detected logs in each step was calculated when APT attacks were detected using GRR alone, and when GRR and Graylog were used together as attack detectors. When calculating the scores, all logs existing in multiple tactics were not treated similarly, and they were reflected in each calculation to assign a higher weight as the number of logs detected for each tactic increased. Here, for the criteria for the successful detection of attack in each step, the sum of the severity scores of the collected logs to three or higher were set. The attack detection time points were compared based on the cumulative log score for each step.

Fig. 5 and Table 5 present the cumulative score of collected logs by step and comparison of the attack detection time points, respectively.

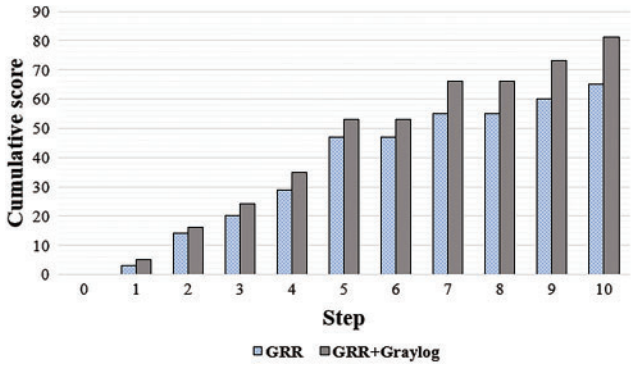

Figure 5: Cumulative score of collected logs by step

Table 5: Comparison of attack detection time points

Level	Threshold	GRR (conventional scheme)	GRR + Graylog (proposed scheme)
Low	40	Step 5	Step 5
Medium	60	Step 9	Step 7
High	80	X	Step 10

As shown in Fig. 5, the cumulative score of the log collected up to Step 10 was 65 points, and that of the proposed model was 81 points. As the threshold is a value specified by the administrator of the detection tool, the detection time points were compared when the threshold of the accumulated score of the collected log was set to low, medium, and high levels. Based on the cumulative severity score of the log in Fig. 5, the detection thresholds were optionally assumed to be 40, 60, and 80, respectively, but they may be assigned different values depending on the administrator's choice. When an alarm for a response is given at the time point of attack detection, the detection time is earlier in the interworking environment of the threat log collector and attack detector. Here, the attack is detected at a time point when the attacker expands the access domain in the case of the low threshold, when

a session is connected to the targeted final victim system in the case of medium threshold and when the attack is performed in the case of high threshold. In a conventional method that uses a single open-source threat detector, detection is not possible at all when the threshold is high. However, in the case of detection in the proposed interworking environment, the APT attack can be detected before its completion, even if the threshold is high. The proposed model produces improved results such that post-event response measures can be prepared by immediately sending an alarm to the analyst at the time point of detection.

4.2.2 Coverage Based on Tactics and Techniques of MITRE ATT&CK

Table 6 details the results of calculating the average coverage of tactics and techniques of MITRE ATT&CK mapped in each step in Table 2. The coverage results of Steps 0–10 during the Carbanak attack scenario are presented. The coverages were expressed by rounding to the first decimal place.

Table 6: Comparison of the average coverage of tactics and techniques

Step	Average coverage of tactics		Average coverage of techniques	
	GRR (conventional scheme)	GRR + Graylog (proposed scheme)	GRR (conventional scheme)	GRR + Graylog (proposed scheme)
0	0	0	0	0
1	20	20	20	20
2	37.5	37.5	60	60
3	37.5	50	30	50
4	70.8	83.3	73.3	93.3
5	75	75	75	75
6	0	0	66.7	66.7
7	36.1	80.6	41.7	75
8	0	0	0	0
9	18.8	43.8	25	41.7
10	41.7	66.7	40	60
Average	30.7	41.5	43.2	54.2

The average coverage of tactics expanded by approximately 10.8% when GRR and Graylog were used simultaneously, compared with when only GRR was used. Generally, at a time point when the access domain is expanded before executing an attack, the proposed model exhibited expanded coverage compared with the method that uses a single threat detector. Furthermore, the coverage was considerably improved at the time of session connection to the final target victim.

The average coverage of techniques increased by approximately 11% in the interworking environment of the open-source threat detector and threat log collector. For the techniques, the coverage improved considerably at the time of access domain expansion and at session connection with the final victim.

4.2.3 Calculation of the Coverage for Each Attack Pattern

The following is the result of calculating the coverage for each type of state machine defined earlier. The Carbanak attack pattern specified in Fig. 2 can be defined as a total of 16 conditions, depending on whether the attacker performs Steps 4, 6, 7, and 8. Condition 1 is a type that goes through all Steps 4, 6, 7, 8, and Conditions 2, 3, 4, and 5 are patterns in which attackers do not go through only Steps 4, 6, 7, and 8. Conditions 6, 7, 8, 9, 10, and 11 are attack types in which attackers do not go through Steps 4, 6 and Steps 4, 8 and Steps 4, 9 and Steps 6, 8 and Steps 6, 9 and Steps 8, 9, respectively. Conditions 12, 13, 14, and 15 are attack types performed by attackers only in Steps 9, 8, 6, and 4, respectively. Finally, Condition 16 is an attack type in which the attackers do not go through all Steps 4, 6, 8, and 9.

Fig. 6 shows the results of calculating the coverage for each attack type. The detection coverage is analyzed on the premise of accurate detection rate based on the ruleset. When GRR was used as a single detector, the average coverage was 32.6%. By contrast, when detection was performed in the interworking environment with Graylog, the average detection coverage of the 16 APT attack patterns was 43.8%, an improvement of approximately 11.2%. The coverage increased by more than the average for the attack types that did not go through Steps 6 or 8 in which additional information of the final victim was acquired. Overall, the detection coverage, which is an indicator of detection accuracy, is improved in all conditions. Thus, the proposed model can be assumed to show an improved detection rate for various APT attack patterns.

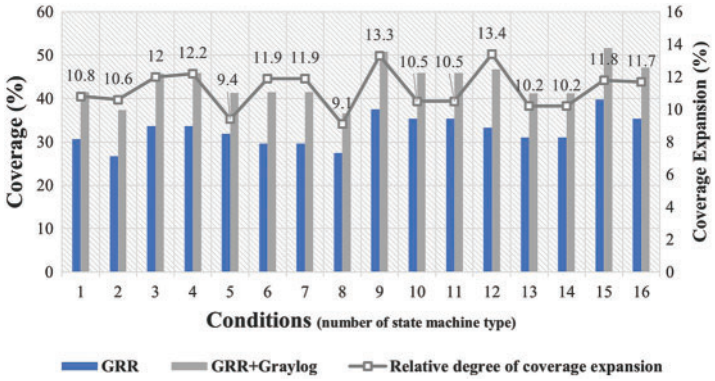

Figure 6: Detection coverage *vs.* conditions

5 Conclusion

In this paper, a method for improving the detection coverage performance of an open-source threat detector was proposed by interworking with an open-source threat log collector. Furthermore, because detection coverage was derived based on various cyberattack techniques defined by MITRE ATT&CK, the proposed model could effectively respond to new attacks in addition to the scenarios considered in this paper. In particular, coverage is examined by summarizing all potential occurrences of APT attack patterns to verify the performance of the proposed model in various attack environments. For example, sensitivity was examined by ensuring that the coverage of the suggested model was upheld when attacks are launched using various patterns — such as uploading and executing shellcode before immediately expanding the scope of the attack by executing Powerview. The consequence of this, as seen in Table 6, was that the performance of our model (which integrated GRR and Graylog under all conditions) was further enhanced. Owing to their nature, open-source tools exhibit lower

performance compared to expensive threat detectors developed on closed platforms; nevertheless, the performance of highly accessible open-source tools could be improved for sufficiently responding to sophisticated attacks that target endpoints. In the case of open-source threat detectors, APT attacks are difficult to detect when a single tool is used. However, if GRR and Graylog are combined, the detection coverage can be expanded by approximately 11%. If an efficient analysis environment is provided based on the severity patterns of logs, then attacks can be detected by blocking connections in advance before the access domain of the attacker expands. The limitation of this study is that the accuracy of the fully automated detection and response was not evaluated. In future work, we will study techniques to detect attackers specifically based on the defined detection rules reinforced by machine learning in various environments. In future work, we will study techniques to detect attackers specifically based on the defined detection rules of an environment where normal and abnormal patterns are mixed. Furthermore, the detection accuracy of attacks will be measured and analyzed by applying the ruleset, and automated response functions will be implemented to block the connections of certain attackers. Subsequently, we will conduct experiments and analyses on detection in various attack environments and evaluate response accuracy and coverage. In addition, we will improve detection rates by introducing metaheuristic algorithms that can counteract various abnormal behavior patterns caused by APT attacks.

Funding Statement: This study is the result of a commissioned research project supported by the affiliated institute of ETRI (No. 2021-026) and partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1F1A1061107), the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (MOTIE) (P0008703, The Competency Development Program for Industry Specialist), and the MSIT under the ICAN (ICT Challenge and Advanced Network of HRD) program [grant number IITP-2022-RS-2022-00156310], supervised by the Institute of Information & Communication Technology Planning and Evaluation (IITP).

Author Contributions: So-Eun Jeon: Conceptualization, Methodology, Validation, Writing – Original Draft. Sun-Jin Lee: Methodology, Formal Analysis, Visualization, Writing – Original Draft. Eun-Young Lee: Software, Investigation, Resources, Writing—Original Draft. Yeon-Ji Lee: Methodology, Software, Investigation. Jung-Hwa Ryu: Investigation, Formal Analysis, Visualization. Jung-Hyun Moon: Investigation, Formal Analysis, Visualization. Sun-Min Yi: Investigation, Formal Analysis, Visualization. Il-Gu Lee: Conceptualization, Validation, Writing—Review & Editing, Supervision, Project administration, Funding acquisition.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Kim, F. A. Alfouzan and H. Kim, "Cyber-attack scoring model based on the offensive cybersecurity framework," *Applied Science*, vol. 11, no. 16, pp. 7738, 2021.
[2] AVTEST, Security Report 2019/2020, The Independent IT-Security Institute, 2020.
[3] A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[4] A. K. Mishra, E. Pilli and M. Govil, "A taxonomy of cloud endpoint forensic tools," in *Advances in Digital Forensics XIV. Digital Forensics. Int. Federation for Information Processing Advances in Information and Communication Technology*, New Delhi, India, vol. 532, pp. 243–261, 2018.
[5] V. Susukailo, I. Opirskyy and S. Vasylyshyn, "Analysis of the attack vectors used by threat actors during the pandemic," in *IEEE 15th Int. Conf. on Computer Sciences and Information Technologies*, Zbarazh, Ukraine, vol. 2, pp. 261–264, 2020.
[6] N. N. A. Sjarif, S. Chuprat, M. N. Mahrin, N. A. Ahmad, A. Ariffin *et al.,* "Endpoint detection and response: Why use machine learning?," in *2019 Int. Conf. on Information and Communication Technology Convergence*, Jeju, South Korea, pp. 283–288, 2019.
[7] The Center for Threat-Informed Defense, "Adversary emulation library," [Online]. Available: https://github.com/center-for-threat-informed-defense/adversary_emulation_library/tree/master/carbanak
[8] A. Berady, V. Tong, G. Guette, C. Bidan and G. Carat, "Modeling the operational phases of APT campaigns," in *Proc. IEEE-Int. Conf. on Computational Science and Computational Intelligence*, Las Vegas, NV, USA, pp. 96–101, 2019.
[9] Z. Ma, Q. Li and X. Meng, "Discovering suspicious APT families through a large-scale domain graph in information-centric IoT," *IEEE Access*, vol. 7, pp. 13917–13926, 2019.
[10] B. Stojanović, K. Hofer-Schmitz and U. Kleb, "APT datasets and attack modeling for automated detection methods: A review," *Computers & Security*, vol. 92, pp. 101734, 2020.
[11] J. Joloudari, M. Haderbadi, A. Mashmool, M. Ghasemigol, S. Band *et al.,* "Early detection of the advanced persistent threat attack using performance analysis of deep learning," *IEEE Access*, vol. 8, pp. 186125–186137, 2020.
[12] N. Mohamed and B. Belaton, "SBI model for the detection of advanced persistent threat based on strange behavior of using credential dumping technique," *IEEE Access*, vol. 9, pp. 42919–42932, 2021.
[13] G. Karantzas and C. Patsakis, "An empirical assessment of endpoint detection and response systems against advanced persistent threats attack vectors," *Journal of Cybersecurity and Privacy*, vol. 1, no. 3, pp. 387–421, 2021.
[14] W. U. Hassan, A. Bates and D. Marino, "Tactical provenance analysis for endpoint detection and response systems," *IEEE Symposium on Security and Privacy*, vol. 1, no. 3, pp. 1172–1189, 2020.
[15] P. Kieseberg, S. Neuner, S. Schrittwieser, M. Schmiedecker and E. Weippl, "Real-time forensics through endpoint visibility," in *Int. Conf. on Digital Forensics and Cyber Crime*, Prague, Czech Republic, vol. 216, pp. 18–32, 2017.
[16] W. Matsuda, M. Fujimoto and T. Mitsunaga, "Real-time detection system against malicious tools by monitoring DLL on client computers," in *IEEE Conf. on Application, Information and Network Security*, Pulau Pinang, Malaysia, pp. 36–41, 2019.
[17] H. Rasheed, A. Hadi and M. Khader, "Threat hunting using GRR rapid response," in *IEEE Int. Conf. on New Trends in Computing Sciences*, Amman, Jordan, pp. 155–160, 2017.
[18] S. Park, S. Yun, S. Jeon, N. Park, H. Shim *et al.,* "Performance evaluation of open-source endpoint detection and response combining google rapid response and osquery for threat detection," *IEEE Access*, vol. 10, pp. 20259–20269, 2022.
[19] Z. Chen, J. Liu, Y. Shen, M. Simsek, B. Kantarci *et al.,* "Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.
[20] S. J. Lee, H. Y. Shim, Y. R. Lee, T. R. Park, S. H. Park *et al.,* "Study on systematic ransomware detection techniques," in *24th Int. Conf. on Advanced Communication Technology (ICACT)*, Pyeongchang, Republic of Korea, pp. 297–301, 2022.
[21] V. Unterfingher, "Ten open-source EDR tools to enhance your cyber-resilience factor," 2020. [Online]. Available: https://heimdalsecurity.com/blog/open-source-edr-tools/
[22] Google, "What is GRR?," 2017. [Online]. Available: https://grr-doc.readthedocs.io/en/v3.2.1/what-is-grr.html
[23] Graylog Team, "About Graylog," [Online]. Available: https://www.graylog.org/about

[24] J. Brown and Y. Pan, "Integrating GRR rapid response with Graylog extended log format," in *Digital Forensics Research Conf.*, Virtual, 2020.
[25] I. Ghafir, K. Kyriakopoulos, S. Lambotharan, F. Aparicio-Navarro, B. AsSadhan *et al.,* "Hidden Markov models and alert correlations for the prediction of advanced persistent threats," *IEEE Access*, vol. 7, pp. 99508–99520, 2019.
[26] The Center for Threat-Informed Defense, "Adversary emulation library," [Online]. Available: https://github.com/center-for-threat-informed-defense/adversary_emulation_library/blob/master/carbanak/Attack_Layers/Carbanak_Scenario1.png
[27] The MITRE Corporation, "MITRE ATT&CK," [Online]. Available: https://attack.mitre.org/versions/v9/
[28] Microsoft, "Trace and event log severity levels," 2016. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/ff604025(v=office.14)
[29] N. E. Park, Y. R. Lee, S. Joo, S. Y. Kim, S. H. Kim *et al.,* "Performance evaluation of a fast and efficient intrusion detection framework for advanced persistent threat-based cyberattacks," *Elsevier Computers and Electrical Engineering*, vol. 105, no. 4, pp. 108548, 2023.
[30] R. Stoleriu, A. Puncioiu and I. Bica, "Cyber attacks detection using open source ELK stack," in *2021 13th Int. Conf. on Electronics, Computers and Artificial Intelligence (ECAI)*, Pitesti, Romania, pp. 1–6, 2021.