Performance Evaluation of Open-Source Endpoint Detection and Response Combining Google Rapid Response and Osquery for Threat Detection

SO-HYUN PARK\textsuperscript{1}, SUN-WOO YUN\textsuperscript{1}, SO-EUN JEON\textsuperscript{2}, NA-EUN PARK\textsuperscript{2}, HYE-YEON SHIM\textsuperscript{2}, YU-RIM LEE\textsuperscript{2}, SUN-JIN LEE\textsuperscript{2}, TAE-RIM PARK\textsuperscript{2}, NA-EUN PARK\textsuperscript{2}, MIN-JIN KANG\textsuperscript{2}, AND IL-GU LEE\textsuperscript{1,2}
\textsuperscript{1}Department of Future Convergence Technology Engineering, Sungshin Women's University, Seoul 02844, South Korea
\textsuperscript{2}Department of Convergence Security Engineering, Sungshin Women’s University, Seoul 02844, South Korea
Corresponding author: Il-Gu Lee (iglee@sungshin.ac.kr)

This work was supported in part by the Commissioned Research Project through the Affiliated Institute of the Electronics and Telecommunications Research Institute (ETRI) under Grant 2020-023, in part by the National Research Foundation of Korea (NRF) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant 2020R1F1A1061107, and in part by the Korea Institute for Advancement of Technology (KIAT) Grant by the Korean Government through MOTIE (The competency development program for industry specialist) under Grant P0008703.

ABSTRACT Detecting the latest advanced persistent threats (APTs) using conventional information protection systems is a challenging task. Although various systems have been employed to detect such attacks, they are limited by their respective operating systems. Furthermore, they are developed as closed platforms and cannot be customized to meet user environments. To overcome these limitations, open-source endpoint detection and response (EDR) techniques are needed. In this study, we construct one that integrates open-source security frameworks combining GRR (Google Rapid Response) and osquery. A threat-detecting case study is conducted to validate the feasibility of the proposed open-source EDR system. Additionally, APT coverage for the proposed EDR system is analyzed using MITRE's Adversarial Tactics, Techniques, and Common Knowledge model. The assessment result shows that APT tactics having high levels of threat detection using non-customized osquery configurations comprise 28.5% of all detections, which is lower than the other response levels. The performance of open-source EDR can be increased by customizing osquery for specific purposes and environments. Open-source EDR combining GRR and osquery has the potential to provide the detection-coverage efficient threat detection system and has the advantage of flexible integration with other applications; it can also be developed for evolving system environments such as cloud and internet of things.

INDEX TERMS Advanced persistent threat, behavior-based detection, cyber-attack, detection criteria, remote live forensics, open source based EDR.

I. INTRODUCTION

Cyber-attack techniques constantly improve, and advanced persistent threats (APTs) cause serious security problems for companies and organizations [1]. It is difficult for existing information protection systems to detect the latest APTs because they attack the targets persistently for a prolonged period using intelligent advanced hacking techniques in a high-density, high-capacity, and high-speed network environment [2]. The most representative method of responding to APTs includes using a cyber kill chain, as first devised as a military security concept in 2011 by Lockheed Martin. It defines cyberattacks in multiple stages, identifies threats to organizational processes in advance, and analyzes, detects, and prevents cyberattacks and intrusions [3]. Its attack stages consist of reconnaissance, weaponization, delivery, exploitation, installation, command and control (C2), and actions-on-objective. Additionally, cyber kill-chain models (e.g., the Mandiant attack lifecycle and the Bryant kill chain) have been developed [4]. Bahrami et al. [5] proposed a taxonomy of APT attacks based on the cyber kill-chain model, according to which tactics, techniques, and procedures for detecting APT attacks were identified. Although the cyber kill chain has emerged as a new framework for responding to APTs, it has
a limitation of only presenting the threats in a single attack stage, and it relies on security solutions, such as intrusion prevention systems, firewalls, and security information and event management tools for responding.

To respond effectively to APTs, behavior-based detection techniques must be applied alongside the kill-chain model [6]. A representative example of signature-based detection is the conventional anti-virus (AV) suite, which defeats malware by substituting their analyzed signatures in the suspected sample. However, signature-based detection has disadvantages of vulnerability to zero-day attacks, a high false-alarm rates, and difficulty responding to attacks that bypass signature detection [7]. By contrast, behavior-based detection techniques analyze the behaviors of malware using artificial intelligence, big data, visualization, and cloud technologies. Refs. [8] and [9] proposed behavior-based malware detection systems for Android. In particular, the system proposed by [8] was based on the fact that new malware is usually a variant of an existing one, whose signatures can be used to detect the new ones based on behaviors and to prevent its future obfuscation and transformation.

However, existing behavior-based detection methods are limited by the operating system (OS) and are developed as closed platforms, making it difficult to customize the user environment. Furthermore, it is impossible to effectively respond to quickly evolving cyberattacks. The larger the network and the more diverse the systems, the more severe the vulnerabilities. Endpoint detection and response (EDR) methods are designed to overcome the disadvantages of behavior-based detection techniques. Because EDRs support actions in various OSs and are developed using open sources, they allow experts worldwide to collaborate and prepare responses faster than the evolution speed of attacks.

Gartner [10] classified threat detection technologies based on the detection point in the network, the endpoint, and the user and classified products, as shown in Table 1. Accordingly, EDRs are being developed into next-generation endpoint threat detection tools by integrating them with endpoint protection platforms. Furthermore, it was forecasted that the total endpoint security market will grow at a compounded annual growth rate of 7.6% from USD 12.8B in 2019 to USD 18.4B in 2024 [11]. These trends of endpoint and EDR markets highlight that the demand and market size for EDR solutions are expected to grow steadily.

EDRs should be used in conjunction with multiple solutions because they must detect and respond to all events occurring at network endpoints. Therefore, EDR solutions typically incorporate endpoint anomaly detection and response technologies alongside existing AV solutions and techniques. Unlike traditional security flows, recent security solutions are built open-source. A representative solution is GRR Rapid Response (GRR), a remote live forensics tool for responding to enterprise intrusions. It also provides digital evidence collection for forensics and incident-response applications. Open-source security solutions, including GRR, have the advantage of extending their capabilities to other open-source solutions and application programming interfaces (APIs), thus enabling flexible utilization. There are several studies related to such open-source security tools, but they only analyze the functionalities of the tools and the feasibility of utilizing them at some specific circumstances. In this study, we propose next-generation endpoint security-threat detection and response methods and evaluate the detection coverage of the suggested EDR system for the first time.

The contributions of this study are as follows:

- An open-source EDR system combining GRR and osquery is evaluated for threat detection, and incident detection experiments are conducted.
- Using MITRE ATT&CK, all available attacks are organized, and the detection coverage of the EDR environments are identified.
- Alongside detection coverage analysis results, development directions are presented for the implementation of open-source EDRs, and their limitations and open issues (future challenges) are discussed.

The remainder of this paper is organized as follows. Section II describes previous studies on open-source EDRs. Section III describes the structure and functions of open-source EDR systems implemented using GRR and osquery. Section IV conducts attack and detection experiments. Section V defines the detection criteria of open-source EDRs and analyzes detection coverage. Section VI presents the discussion and future research directions. Finally, section VII concludes the study.

## II. RELATED WORKS

In this section, we summarize the literatures related to digital evidence collection and incident response using open-source-based digital forensics and incident response (DFIR) tools and the security solutions. Like this study, several others have attempted to detect, respond to, and analyze cyber-security limitations using GRR and osquery. Table 2 describes the related works.

Reference [12] created memory corruption and persistent attacks using Metasploit on GRR clients and responded to them using GRR’s hunt functionality (i.e., GRRScanMemoryHunt, NetworkStatusHunt, GRRRegistryFinderHunt). Using GRR’s cronjob, hunt tasks were scheduled, and memory, network status, and Windows registry-key analysis were performed, resulting in a well-documented analysis of the limitations of slow detection caused by the hunt cycle taking
a long time (~5–10 min). During this period, attackers had enough time to complete their attacks and cover their tracks. Furthermore, [13] described the system structure and functionality of GRR, osquery, and Mozilla Investigator, comparing their performances by establishing representative features that were successfully handled. Although both studies suggest the limitations of EDR, there is a limitation in that they do not specifically address the solution. In addition, although attempts have been made to identify the core functions of EDR, there is insufficient research on how to respond to overall cyber-attacks.

References [14]–[17] applied a GRR to several domains to collect data and attempted forensic analysis. Ref. [14] used GRR to respond to cyber threats in a healthcare system, displaying the benefits of operating the system simultaneously with remote forensics. Furthermore, they rendered the tool compatible with multiple frameworks, owing to its open-source nature. However, there was an issue that an additional security mechanism is needed for GRR to safely handle sensitive information in the medical system. Containers have also been frequently used to increase the convenience and efficiency of software engineers. Refs. [15]–[17] highlighted the need for digital forensics for web servers operating in containers (e.g., Docker and Kubernetes) and used some to respond to distributed denial-of-service attacks. In these studies, the method of recognizing a DDoS attack by checking the web server log and IP address was used. However, it is not different from the existing network security techniques and no automation attempts were made.

TABLE 2. Summary of related works.

| Refs. | Contribution | Limitation |
|-------|--------------|------------|
| [12]  | Responded to memory corruption and persistent attacks dealing with instead of respond to using GRR hunt functionality. | Only a few specific attacks were resolved. |
| [13]  | Described and compared the system structure and functionality of GRR, osquery, and Mozilla Investigator. | Limitations in resolving the attacks were not addressed. |
| [14]–[17] | Applied GRR to several domains to collect data and attempt forensic analysis. | The method used was not different from the existing network security techniques, and no automation attempts were made. |
| [18]–[19] | Integrated GRR with other analytical tools. | Routing method was used instead of using normal methods to access logs. |
| [20]  | Analyzed and resolved the database limitations of GRR. | Additional studies are needed in processing, transmitting, and utilizing the collected data. |

Furthermore, [20] analyzed and resolved the database limitations of GRR. Log data were stored in a database for incident response and forensic evidence collection, but the method had limitations of increased data with larger-scale services, resulting in resource shortages. Ref. [20] proposed a distributed data repository to overcome the limitations of GRR storage systems, improving scalability, processing speeds, and efficiency. However, it is necessary to study not only the data collection stage, but also the performance of processing, transmitting, and utilizing the collected data. When using the collected log data, additional studies such as performance comparison between the existing method and the distributed storage method can be conducted. In this paper, in conjunction with EDR and Kafka, we propose a way to stream the collected log data and utilize it.

Most of the existing studies dealt with only a few attacks in specific situations and conditions, and were limited to basic explanations of the main functions and limitations of each EDR tool. However, in order to respond to APT attacks that are becoming more intelligent day by day, it is necessary to study the EDR system that analyzes attacks at each stage of the APT scenario like a cyber kill chain and systematically responds to them. In this work, we further seek to construct a responsive EDR system with endpoint anomaly detection by processing the collected endpoint data used in extant studies, wherein GRR and osquery are used for DFIR. Also, the performance of proposed EDR system is evaluated via analyzing the detection coverage for all APT stages.

III. OPEN-SOURCE EDR SYSTEMS OVERVIEW

This section describes GRR and osquery systems and introduces the EDR environments that combines them.

A. GRR

GRR is an incident-response framework for remote live forensics and includes a client and a server [21]. GRR clients are distributed to systems that need to be detected, and they periodically poll the server for work, referring to a client action that includes file finder, memory, network, OS, and osquery. The GRR client and server communicate via the hypertext transfer protocol (HTTP) by default, and each communication is encrypted using AES (Advanced Encryption Standard) 256.

The GRR server comprises a front-end server, a worker, and a user-interface, and its core functions include flow and hunt. The GRR server uses Flows, a type of state machine, to solve resource problems. A Flow is a core entity of the server, and it calls client actions. When the GRR server first enters its flow state, it requires a client action for the state. While waiting for a response from the client, the server initializes all resources. When a response arrives, it fetches the appropriate resource and runs the flow state. This way, the GRR can resolve resource-hogging issues. Flows can be done on thousands of client machines, which is called a hunt.
"schedule": {
  "file_events": {
    "query": "SELECT * FROM file_events;",
    "interval": 300
  }
}

FIGURE 1. Example of osqueryd scheduling.

Hunt specifies the flow components for which machines to run the flow on.

GRR stores forensic data collected from clients and abstracts and represents them using a virtual file system. Hence, analyzers can identify the client file system. The GRR artifact is a function that collects and manages data generated while using the OS or applications. It can be conveniently used in conjunction with other tools because the information collected from the host is stored in an external storage system in YAML (YAML: Ain’t Markup Language) format, a human-friendly data serialization standard for all programming languages. GRR-specific or private artifacts are stored locally. For other functions, GRR supports the Python library (i.e., grr_api_client) for automation and uses PowerShell for automation and scripting. Furthermore, it provides a function that interfaces osquery during the flow of the GRR and obtains client information in a structured query language (SQL) format for more effective analysis.

B. OSQUERY

Osquery is an operating system instrumentation framework that abstracts and displays information for analysis and monitoring in an SQL table and can be used with MacOS, FreeBSD, Linux, and Windows. Since October 2020, it supports 264 basic schema and provides system information (e.g., processes, CPU, disc, device, memory, kernel, files, Wi-Fi, and network information etc.) and information related to YARA malware research and detection, Docker containers, and Azure cloud services. New schema can be defined to YARA malware research and detection, Docker containers, and Azure cloud services. Osquery provides a function that interfaces osquery during the flow of the GRR and obtains client information in a structured query language (SQL) format for more effective analysis.

FIGURE 2. EDR system configuration.

ID. Consumers can receive a schema ID from the schema registry, search the corresponding schema, and use it by reverse-serializing it [23].

C. EDR SYSTEM ENVIRONMENT

The experimental environment of open-source EDR was set up by building a GRR server and client as virtual machines using VMware. First, we verified whether the detected data were properly transmitted to the EDR in an experimental environment consisting of the GRR server, client, and Kafka. The GRR server does not provide an osquery agent installer, and a process is required to install osquery’s official source in the client machine. When osquery is installed on the client machine and placed on a path that can be accessed by GRR, GRR and osquery can be used together.

Fig. 2 shows the EDR experimental setup comprising GRR, osquery, and Kafka. The GRR server is connected to the GRR client and uses the HTTP protocol for communication. The GRR client in which osquery has been installed sends query results to the GRR server according to its request. MySQL, the database of the GRR server, is linked to Kafka and can stream the sent detection results.

The hunt function of GRR can cause performance degradation of running clients and GRR systems because it results in an extensive load on the system [24]. This causes the hunt to expand its coverage, making it difficult to use for live forensics. When utilizing the hunt function to solve this problem, limits on resource usage should be set (limited). To do so, data can be accumulated in external systems, and forensics can be performed separately. Kafka can be used to transmit data from internal to external systems and to store, manage, and investigate them externally [25]. Kafka is a distributed streaming platform with proven big-data processing performance. In this paper, to resolve the abovementioned performance and scalability problems, the Kafka-based open-source EDR is studied.

IV. ATTACK AND DETECTION EXPERIMENTS

A. ATTACK SCENARIO AND EXPERIMENT

For attack detection experiments with open-source EDR, this study used the MITRE ATT&CK framework of hostile tactics and techniques used for classification and risk evaluation at
an organizational level. It provides organizational security awareness, helps identify gaps in defense, and prioritizes risks [26]. The enterprise ATT&CK framework used in this study is based on the general attack procedure, which refers to actions performed from the intrusion stage to the goal achievement stage. This framework sequentially describes 12 attack stages (i.e., initial access, execution, persistence, privilege scaling, defense evasion, credential access, discovery, lateral movement, collection, C2, exfiltration, and impact), which include a total of 184 techniques.

The attack scenario for penetration testing consists of initial access, execution, C2, and impact stages of ATT&CK. For penetration testing, we used Metasploit, an exploitation and vulnerability validation tool that assists penetration testing, and the open-source ransomware technique, RAASNet. Wine was also used to run the Windows.exe file in Linux OS. The attack scenario was set based on the APT. First, a Metasploit payload disguised as a normal file is generated and moved to a universal serial-bus (USB) connected to the victim’s personal computer (PC). When a malicious payload is stored and executed on a victim’s PC via USB, the attacker’s C2 server and the victim’s PC become connected. Hence, abnormal symptoms detected in the network traffic can be verified and responded to. Furthermore, the victim’s PC is infected by the RAASNet ransomware, and it can be responded to by detecting the characteristics encrypted in a file with a specific
extension (demon), and the file size and time property are changed.

Fig. 3 shows the attack-and-response environment of the experiment, which comprises a GRR server, a client, and an attacker. It shows the attack stages of initial access, execution, C2 corresponding to the attack scenario, and the flow of the attack. The GRR server and client respond to attacks while communicating via HTTP. Table 3 shows the conditions of the experiment.

1) INITIAL ACCESS
The initial access-stage attack consists of techniques related to the attacker’s network access attempts. The attacker can access the victim’s PC first by inserting malicious exploit code or an attachment file into the web browser, e-mail, or acquiring network access permission. The attacker can also attempt malicious actions by copying malware to a mobile device and inserting it into the system. In this experiment, malware was stored on a USB mobile storage device and inserted into the victim’s system. “SELECT * FROM usb devices;” was used as the query statement to respond to the initial access stage. The information regarding the USB devices connected to the PC (class, model (modelname), model_id, protocol, removable, serial, subclass, usb_address, usb_port, vendor, vendor_id, version) can be known, and traces of the USB connection at the time of initial attack can be proven. In particular, the ID and serial number in the USB device information can be used as critical information to detect unauthorized USBs in addition to those registered by the organization. Furthermore, the information of file_events in a table related to file-integrity monitoring can be output using the query, “SELECT * FROM file_events;” to display the states of files like “/media/account/USBname/%”;”

2) EXECUTION
The Metaspliot payload injected via the USB in this experiment was a malicious execution file disguised as a normal file, and it bypassed security and AV programs. Among the detailed attack techniques of malware, user execution occurs when the user opens the file. When the payload is executed, a session is then formed between the C2 server and the victim’s PC, and network traffic is generated. Because the experimental environment of the victim’s PC was Ubuntu OS, Wine was used to run the exe file.

3) C2
A Trojan attack induces the internal system to voluntarily open a port to enable communication with external systems. In this case, it opened a specific port for C2-server communications. When the network traffic is detected, it can be judged as an abnormal symptom, and hacking can be suspected. In this case, it was verified that a connection between the attacker and the victim’s system was initiated, owing to verification via a query designed to detect suspicious outbound network activities: “SELECT s.pid, p.name, local_address, remote_address, family, protocol, local_port, remote_port FROM process_open_sockets s JOIN processes p on s.pid = p.pid WHERE remote_port not in (80, 443) and family = 2;”

4) IMPACT
This step is the final step of APT attack, and it interferes with the availability of systems, services, and resources, or it damages integrity. In this experiment, the victim’s PC was infected with ransomware, thereby damaging the availability and integrity of the target folder. The characteristics that can be used to check the behaviors of the ransomware are file extension, timestamp (time information), and change in file size. The properties of the folder, file extension, and size, and timestamp can be checked using the query, “SELECT * FROM file WHERE path LIKE ‘/home/user_name/ransomware-infected folder/%’;” Additionally, we can use a query that only shows information regarding a specific extension, “SELECT * FROM file WHERE path LIKE ‘/home/user_name/ransomware-infected folder%%extension’;” or one that determines changes in time values by ransomware infection by checking atime, mtime, and ctime via “SELECT filename, atime, mtime, ctime FROM file WHERE path LIKE ‘/home/user_name/ransomware-infected folder%’;” Furthermore, the information of file_events in a table related to file-integrity monitoring can be output using the query, “SELECT * FROM file_events;” to display the states of files as created, deleted, or updated. Table 4 shows the summary of the attack and detection experiment methods. It reflects the stages of the MITRE ATT&CK, the attack techniques, and the osquery queries for all attacks. Queries that show the most representative characteristics of each attack were selected for this experiment, and the attacks were detected using them.

B. RESULT OF DETECTION SCENARIO
As a result of conducting the attack-and-response experiment assuming an APT attack scenario based on MITRE ATT&CK, various attacks, including the use of an unauthorized USB, a Trojan horse, and ransomware, were detected using osquery input query statements. GRR provided a client API that enabled Python scripting and automation, and osquery provided the osqueryd daemon, which recorded logs when events were detected. This was used to set response schedules to activate the EDR system according to the...

| Components          | Used version     |
|---------------------|------------------|
| GRR server, client OS | Ubuntu 18.04.4 LTS |
| GRR server, client  | 3.4.0-1          |
| osquery              | 4.3.0            |
| Metasploit           | Kali-linux-2020.2- amd64 |
| RAASNet              | https://github.com/leonov24/RAASNet |
| Wine                 | 3.0              |
established period. Therefore, the occurrence of events in a specific path or file can be detected in real time, and any abnormal symptoms can be responded to immediately by scheduling osqueryd. Through this process, automated detection and response tools that detect changes via specific events and perform investigations of corresponding artifacts are made feasible. Furthermore, the detected data can be transmitted via Kafka, one of osquery’s logger plugins, to be utilized for analysis and response. The results of this detection experiment show that several open-source security tools can be interlocked to form an EDR environment to detect, analyze, and respond to endpoint security incidents.

V. PERFORMANCE EVALUATION
This section defines the detection criteria of open-source EDR, tests its detection coverage, and analyzes the results. Additionally, a developmental direction for improving detection performance is presented.

A. DEFINITION OF DETECTION CRITERIA
In our research environment, the security-incident detection criteria were established by organizing the information and query-statement requirements required to detect MITRE ATT&CKs. For example, for drive-by-compromise attack techniques during an initial attack tactic, because the attack can be detected through website-access and script execution logs, the query statement to verify the log becomes a requirement for detection. Additionally, the valid-account attack technique can be detected from user account information, log-change permissions, etc., which require query statements to verify that new unauthorized local accounts have been created. It also requires queries to check for unusual changes in privilege escalation alongside all appliances and applications for default credentials and secure-shell protocol keys.

TABLE 4. Summary of the attack and detect experiment methods.

| Attack          | Stage              | Technique            | Queries                                                                 |
|-----------------|--------------------|----------------------|-------------------------------------------------------------------------|
| Unauthorized USB| Initial access     | Replication through removable media | SELECT * FROM usb_devices;                                                |
|                 |                    |                      | SELECT * FROM file WHERE path like "\media/account/USB name%";          |
| Trojan horse    | Command and control| Dynamic resolution   | SELECT s.pid, p.name, local_address, remote_address, family, protocol, local_port, remote_port FROM process_open_sockets s join processes p on s.pid=p.pid WHERE remote_port not in (80, 443) and family=2; |
| Ransomware      | Impact             | Data destruction     | SELECT * FROM file WHERE path LIKE "/home/user name/ransomware-infected folder%/"; |
|                 |                    |                      | SELECT * FROM file WHERE path LIKE "/home/user name/ransomware-infected folder%/extension"; |
|                 |                    |                      | SELECT filename, atime, mtime, ctime FROM file WHERE path LIKE "/home/user name/ransomware-infected folder%/"; |

TABLE 5. Criteria for detection-level classification.

| Level     | Number of queries satisfying requirements | Number of requirements per technique |
|-----------|------------------------------------------|-------------------------------------|
| High      | 70% or higher                            | higher                              |
| Medium    | 40% or higher                            | Less than 70%                       |
| Low       | Less than 40%                            | 40%                                 |

B. ANALYSIS OF DETECTION COVERAGE
For detection coverage analysis, we investigated and developed query statements to collect a total of 38 queries from osquery GitHub1 opened by Facebook, osquery query packs,2 osquery-configuration GitHub,3 osquery document,4 and other related research and publication data websites.5 A total of 691 were collected and developed, and 381 were summarized as a result of merging and deleting duplicate query statements (160 from osquery GitHub, 136 from the osquery query pack, 48 from websites, 31 from related research data, four from osquery documents, and two custom produced according to the collection routes).6 The queries used in this experiment are all available on github. Based on the ATT&CK framework, the distribution of detection data was generally proportional to the number of techniques at each stage.

To define detection coverage, the 381 query statements were analyzed to see whether they satisfied the requirements of detection. Then, the detection level of the GRR was
TABLE 6. ATT&CK matrix and weight assignment standards based on the teach model.

| Model name          | Description                                                                 | Weight |
|---------------------|-----------------------------------------------------------------------------|--------|
| Techniques Only     | Not abused actually. If other techniques must be used to actually run it    | 5      |
|                     | For example, Graphical User Interface, User Execution, Query Registry        |        |
| Exploitable to Anyone | Easy attacks that anyone can exploit                                          | 4      |
|                     | POC malware, script, or other tools are not required to attack                |        |
|                     | For example, Accessibility Features, Registry Run Keys/Start Folder, Image File Execution Options, PowerShell, Scripting |        |
| Additional Steps Required | Tools such as Metasploit or POC script are required for attacks             | 3      |
|                     | More advanced attack than that found in green Capture, Pass the Hash         |        |
| Cost Prohibitive    | Additional infrastructure must be built for attack                           | 2      |
|                     | Some attacks are easy, and some attacks are advanced                         |        |
|                     | For example, Web Shell, Taint Shared-Content, Shared Webroot                 |        |
| Hard                | User-defined dll/exe may be required                                         | 1      |
|                     | Requires in-depth knowledge of the OS for attack                            |        |
|                     | For example, Process Injection, Bootkit/Rootkit, System/Component Firmware   |        |

identified by converting the ratio of the number of query statements that satisfied the requirements to a percentage.

The response levels for the attack techniques are outlined in Table 5. Depending on the percentage that show satisfaction, fewer than 40 % were classified as low, 40 % or higher and less than 70 % as medium, and 70 % or higher as high. In addition, the importance of each technique was calculated alongside the classification criteria for the response levels given in Table 5, and a high-level analysis of detection coverage was performed. To this end, weights were calculated based on the Teach model, which defines the difficulty of attack for each technique opened by ATT&CK. We assumed that the lower the attack difficulty, the more frequent the attack and the resulting damage appears; hence, a higher weight was given to them, as shown in Table 6. The detection coverage was calculated by applying the following equation according to the response level and attack difficulty; it was converted to a percentage using Eq. (1):

\[
\sum_{i=1}^{5} (Weight) \times (Number\ of\ techniques).
\]  

C. ANALYSIS RESULTS

The analysis result of the detection coverage based on ATT&CK is shown in Fig. 5. In the open-source EDR environment of GRR, the ratio of techniques having high response levels was found to be 28.5 %, the ratio of techniques having medium response levels was 35.1 %, and the ratio of techniques having low response levels was 36.4 %. Thus, the ratio of the high response level was lower than that of the others.

The reason for low detection coverage is the lack of query statements for attack detection, which can improve coverage and performance through query statements custom-made for the environment and purpose, rather than basic query statements. Because osquery is based on relational databases, it can obtain the necessary information flexibly by joining multiple schema. For example, in Table 4, a query statement for detecting Trojan attacks was generated by joining the process_open_sockets schema with the processes schema.

To maximize the utilization of GRR and osquery-based open-source EDR, such customized settings for user environments are required. Furthermore, we conducted in-depth research on the detection level and the ratio of detection data at each attack stage and the stages having low detection levels and summarized the performances and development directions of open-source EDR. Fig. 6 analyzes the possibility of response to the attack techniques in each of the 12 attack stages of the MITRE ATT&CK, which represents the detection level of each attack stage. The average number of query statements for threat detection in each attack stage was 52.25. The top-four stages with low response levels where the number of queries is lower than the average were initial access, execution, credential access, and lateral movement, and the numbers of queries in each stage are 17, 26, 29, and 17, respectively.

Fig. 7 lists the ratios of detection data of each attack stage in ascending order, and the ratio was determined by the number of query statements for threat detection to the total number of query statements. The average ratio of the detection data was 5 %. The ratios were equal to or lower than the average ratios in the four stages having insufficient response: 3 % for initial access and lateral movement, 4 % for execution, and 5 % for credential access.

Table 7 lists the techniques having high importance and low detection levels among the total 184 techniques.
### TABLE 7. Techniques with low detection level and high importance.

| Attack stage          | Techniques                                      |
|-----------------------|-------------------------------------------------|
| Initial Access        | Supply Chain Compromise                         |
|                       | Trusted Relationship                            |
| Execution             | Scheduled Task/Job                              |
|                       | Software Deployment Tools                       |
| Privilege Escalation  | Group Policy Modification                       |
| Defense Evasion       | File and Directory Permissions Modification     |
|                       | Group Policy Modification                       |
|                       | Obfuscated Files or Information                  |
| Credential Access     | Network Sniffing                                |
|                       | Steal Web session Cookie                        |
| Discovery             | Domain Trust Discovery                          |
|                       | File and Directory Discovery                    |
|                       | Network Share Discovery                         |
|                       | Network Sniffing                                |
|                       | Query Registry                                  |
|                       | System Network Configuration Discovery           |
|                       | System Owner/User Discovery                     |
|                       | System Time Discovery                           |
| Lateral Movement      | Lateral Tool Transfer                           |
|                       | Remote Services                                 |
|                       | Replication Through Removable Media             |
| Collection            | Archive Collected Data                          |
|                       | Clipboard Data                                  |
|                       | Data from Cloud Storage Object                  |
|                       | Data from Information Repositories              |
|                       | Data from Local System                          |
| Command and Control   | Dynamic Resolution                              |
| Impact                | Defacement                                      |
|                       | Service Stop                                    |

The weights were set as four or higher by referring to Table 6 for importance. Among all techniques, the number of techniques having low detection levels was 64, and techniques having high importance among these was 29, which accounted for 45%. This is approximately half and corresponds to approximately 15.7% of all techniques. In other words, these attack techniques belong to the stage having a higher possibility of attack than other attack techniques, owing to the lower difficulty of attack. However, the detection level was low. In particular, the attack techniques corresponding to the four stages having the lowest detection level in Fig. 7 (i.e., initial access, lateral movement, execution, and credential access) had the lowest relative detection levels in all attacks. Therefore, to improve performance and coverage of the open-source EDR system, additional research and development of technology that can respond to the stages in Table 7 and detailed attacks based on the above four stages are required.

### VI. DISCUSSION AND FUTURE WORK

The open-source EDR is convenient and effective for managing multiple devices, and the source code can be freely modified and improved to support various applications and OSs. Therefore, it is superior to conventional AV and forensic tools in terms of scalability, efficiency, and cost. In this paper, the detection coverage of the proposed EDR system combining GRR and osquery was evaluated, but more effective tools can be combined and improve the detecting performance.

Remote monitoring can be effectively performed using an open-source EDR in the cloud, across the internet of things (IoT), and in large-scale service environments. Additionally, using the collected detection data to respond to attacks or for digital forensics can be considered.

However, current digital forensics procedures require data collection while blocking all network connections to the system, from which evidence may be collected with integrity. Because system data can be collected remotely using EDRs, and the integrity can be verified using hash functions, digital forensics procedures and regulations must be updated in accordance to attack trends. Furthermore, privacy and critical information leakage can occur because real-time logs are collected while confidential information is processed [14]. Hence, information protection at the management server is needed for processing the detection data, responding to attacks, and conducting remote forensics using the open-source EDR.

Furthermore, cloud and IoT usage increases with the acceleration of digital transformation and the development of wireless communication and network technology. Cloud and
virtual system environments require data integrity and digital evidence analysis solutions. IoT has security vulnerability problems as a result of its pursuit of low price, light weight, and unburdened performance. Open-source EDR must evolve to meet these needs. As a future work, the proposed framework will be applied to and evaluated in environments having large-scale IoT devices and cloud environments.

VII. CONCLUSION
The open-source EDR is a cost-effective security tool with high expected values in terms of flexibility, utilization, and scalability, and it can be used in next-generation digital platforms that become hyper-connected, hyper-intelligent, and globally scaled. In this study, attack detection and coverage analysis were possible for all APT attack stages according to MITRE attack through open source EDR for the first time. A few stages showed a low detection rate due to insufficient query settings to detect detailed attacks of each stage. In-depth performance evaluation was conducted using Teach model, and attacks which had high importance and low detection level were analyzed. To ensure the coverage of GRR and open-source EDRs, appropriate query statements tailored to the required environment and conditions must be created, especially those based on low-level and high-importance attacks. Also, other effective open-source-based tools can be used to increase the performance of EDR system. As a future work, other opensource tools can be compared for their performance using the evaluation framework of this study.

REFERENCES
[1] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, “A risk management approach to defending against the advanced persistent threat,” IEEE Trans. Dependable Secure Comput., vol. 17, no. 6, pp. 1163–1172, Nov. 2020.
[2] S. Chandel, M. Yan, S. Chen, H. Jiang, and T.-Y. Ni, “Threat intelligence sharing community: A countermeasure against advanced persistent threat,” in Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR), San Jose, CA, USA, Mar. 2019, pp. 353–359.
[3] M. S. Khan, S. Siddiqui, and K. Ferens, “A cognitive and concurrent cyber kill chain model,” in Computer and Network Security Essentials. Cham, Switzerland: Springer, 2018, pp. 585–602.
[4] B. D. Bryant and H. Saiedian, “A novel kill-chain framework for remote system log analysis with SIEM software,” Comput. Secur., vol. 67, pp. 198–210, Jun. 2017.
[5] P. N. Bahrami, A. Dehghantanha, T. Dargahi, R. M. Parizi, K. K. R. Choo, and H. H. Javadi, “Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures,” J. Inf. Process. Syst., vol. 15, no. 4, pp. 865–889, 2019.
[6] I. Tarnowski, “How to use cyber kill chain model to build cybersecurity?” Eue J. Higher Educ. IT, Feb./Nov. 2017. [Online]. Available: https://www.eunis.org/download/TNC2017/TNC17-IreneuszTarnowski-cybersecurity.pdf
[7] H. S. Galal, Y. B. Malady, and M. A. Atiea, “Behavior-based features model for Malware detection,” J. Comput. Viro!. Hacking Techn., vol. 12, no. 2, pp. 59–67, May 2016.
[8] M. Sun, X. Li, J. C. S. Lui, R. T. B. Ma, and Z. Liang, “Monet: A user-oriented behavior-based malware variants detection system for Android,” IEEE Trans. Inf. Forensics Security, vol. 12, no. 5, pp. 1103–1112, May 2016.
[9] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, “MADAM: Effective and efficient behavior-based Android malware detection and prevention,” IEEE Trans. Dependable Secure Comput., vol. 15, no. 1, pp. 83–97, Jun. 2018.
[10] T. Bussa, A. Litan, and E. Ahlm. (2016). The Fast-Evolving State of Security Analytics. [Online]. Available: https://www.gartner.com/doc/3274217/fastevolving-state-security-analytics
[11] A. Litan. (2020). Morphing EDR Market Grows to $1.5 Billion in 2020. [Online]. Available: https://blogs.gartner.com/avirah-litan/2020/03/15/morphing-ede-market-grows-to-1-5-billion-in-2020
[12] H. Rasheed, A. Hadi, and M. Khader, “Threat hunting using GRR rapid response,” in Proc. Int. Conf. New Trends Comput. Sci. (ICTCS), Amman, Jordan, Oct. 2017, pp. 155–160.
[13] P. Kieseberg, S. Neuner, S. Schrttiewiser, M. Schmiedecker, and E. Weippl, “Real-time forensics through endpoint visibility,” in Proc. Int. Conf. Dig. Forens. Cyber Crime, Cham, Switzerland: Springer, 2017, pp. 18–32.
[14] S. Acharya, W. Glenn, and M. Carr, “A GRReat framework for incident response in healthcare,” in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Washington, DC, USA, Nov. 2015, pp. 776–778.
[15] I. Riadi and A. Sugandi, “Forensic analysis of Docker swarm cluster using grr rapid response framework,” Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 2, pp. 459–466, 2019.
[16] I. Riadi, R. Umar, and A. Sugandi, “Web forensic on container services using grr rapid response framework,” Sci. J. Informat., vol. 7, no. 1, pp. 33–42, Jun. 2020.
[17] I. Riadi, R. Umar, and A. Sugandi, “Web forensic on kubernetes cluster services using grr rapid response framework,” Int. J. Sci. Technol. Res., vol. 9, no. 1, pp. 356–358, 2020.
[18] S. Haas, R. Sommer, and M. Fischer, “Zeeek-osity: Host-network correlation for advanced monitoring and intrusion detection,” in Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection, Sep. 2020, pp. 248–262.
[19] J. Ming and M. Xie, “Remote live forensics for Android devices,” presented at the IEEE Conf. Commun. Netw. Secur. (CNS), Philadelphia, PA, USA, Oct. 2016.
[20] F. Cruz, A. Moser, and M. Cohen, “A scalable file based data store for forensic analysis,” Digit. Invest., vol. 12, pp. S90–S101, Mar. 2015.
[21] M. I. Cohen, D. Bilby, and G. Caronni, “Distributed forensics and incident response in the enterprise,” Digit. Invest., vol. 8, pp. S101–S110, Aug. 2011.
[22] Apache Software Foundation. (2017). Introduction. [Online]. Available: https://gafka.apache.org/intro
[23] Confluent. (2021). Schema Registry Overview. [Online]. Available: https://docs.confluent.io/current/schema-registry/index.html
[24] GRR Team. (2017). Hunt Limits DOs and DONTs. [Online]. Available: https://grr-doc.readthedocs.io/en/v3.2.1/investigating-with-grr/hunts/limits.html
[25] D. Gonzales, Z. Winkelman, T. Tran, R. Sanchez, D. Woods, and J. Hollywood, “Digital forensics compute cluster: A high speed distributed computing capability for digital forensics,” Int. J. Comput. Inf. Eng., vol. 11, no. 8, pp. 960–967, 2017.
[26] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, “Mitre att&ck: Design and philosophy,” MITRE Corp., McLean, VA, USA. Tech. Rep., 2018.
SO-EUN JEON is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, embedded IoT systems, and networks with an emphasis on information security.

NA-EUN PARK is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, malware detection, convergence security, and system security.

HYE-YEON SHIM is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include deep learning, malware detection with artificial intelligence, programming, and malicious behavior analysis.

YU-RIM LEE is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, malware detection, image security, and video security.

SUN-JIN LEE is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, malware detection, image security, and video security.

TAE-RIM PARK is currently pursuing the integrated B.S./M.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, network security, malware detection, cloud computing, and the Internet of Things.

NA-YEON SHIN is currently pursuing the B.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence, block-chain, convergence security, and privacy.

MIN-JIN KANG is currently pursuing the B.S. degree with the Department of Convergence Security Engineering, Sungshin Women’s University, Seoul, South Korea. Her current research interests include artificial intelligence-based malware analysis and detection, and artificial intelligence-based digital forensic.

IL-GU LEE received the B.S. degree in electrical engineering from Sogang University, Seoul, South Korea, in 2003, the M.S. degree from the Department of Information and Communications Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2005, and the Ph.D. degree from the Computer Science and Engineering Department, Graduate School of Information Security, KAIST, in 2016.

He is currently a Professor with the Department of Convergence Security Engineering, Sungshin Women’s University (SWU), Seoul. Before joining SWU in March 2017, he was with the Electronics and Telecommunications Research Institute (ETRI) as a Senior Researcher from 2005 to 2017. He served as a Principal Architect and a Project Leader with Newratek, South Korea; and Newracom, USA; from 2014 to 2017. He has authored/coauthored more than 90 technical articles in the areas of information security, wireless networks, and communications, and holds about 160 patents. His current research interests include wireless/mobile networks with an emphasis on information security, networks, and wireless circuit and systems.

Dr. Lee is also an Active Participant of and a Contributor to the IEEE 802.11 WLAN Standardization Committee.

* * *