

Ransomware Detection Using Open-source Tools

Sun-Jin Lee, Hye-Yeon Shim, Yu-Rim Lee, Tae-Rim Park, Il-Gu Lee

Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea

{220214013, 220214012, 220214014, 220214011, iglee}@sungshin.ac.kr

Abstract—The recent development of new and variant malicious codes, and the increase in cyberattacks in the form of intelligent Advanced Persistent Threat (APT), has led to rapidly increasing levels of damage. In particular, in the case of ransomware, the damage per attack is large, because ransomware uses a network propagation method, by which each attack can infect multiple victims. As ransomware as a service (RaaS) has increased recently, even people without the capacity to develop malicious code have become able to attack via ransomware. In this study, we built and experimented with a framework that detects ransomware in network and system environments using open-source tools. This study showed through analysis and experiments that open-source tools can quickly identify and respond immediately to APT attacks.

Keywords—Open-source, Endpoint Detection and Response (EDR), Google Rapid Response, Open-source HIDS SECURITY (OSSEC), osquery, Ransomware Detection

I. INTRODUCTION

Due to the prolonged COVID-19 pandemic, remote work has become commonplace, and the number of industries using the Internet of Things (IoT) has increased, expanding the attackable area. Among the potential modes of attack, ransomware is widely used in crimes, because it can attack multiple people simultaneously through networks. Ransomware—a portmanteau of Ransom and Software—penetrates the system to be attacked, encrypts data, and then demands money from the victims in return for decryption. Attack types include penetrating the system to encrypt data, stealing confidential information, and blocking

critical systems [1]. Starting with the infection of > 230,000 computers across 150 countries through encryption by "WannaCry" in 2017, ransomware has increased its attack success rate by utilizing new and modified malware and obfuscation techniques. Even attackers who cannot develop their own malware have recently used ransomware as their main means of attack, by licensing malware from hackers and sharing the profits obtained [2, 3]. According to Statista, the number of ransomware outbreaks worldwide in 2020 was around 340 million, up 61.8% from 2019 [4]. According to the Cyber Attack Trends 2021 Mid-year Report released by Checkpoint, the number of ransomware attacks in the first half of 2021 increased by about 93% over six months, and ransomware damage is rising rapidly [5]. Therefore, research into systems which can respond efficiently to cyberattacks is needed to respond to and minimize the damage caused by ransomware. The use of digital forensics technology is important to quickly detect and efficiently respond to intelligent and advanced cyberattacks. Among the digital forensic technologies, research into Endpoint Detection and Response (EDR), an approach which involves the analysis of a client's system event log to identify attacker intentions and respond immediately to attacks is underway [6]. An EDR tool connects a server to a client, allowing them to rapidly obtain client information from the server, and then analyze and respond to this information. This study used open-source EDR tools to analyze in real-time whether a client connected to an EDR server was infected with ransomware, and to check whether an attack could be detected based on this information. The experiments used File Finder [7], one of the flows of Google Rapid Response (GRR), an open-source EDR tool, Query's file-related query statement [8], and Open-source HIDS Security (OSSEC)'s integrity monitoring function [9] to analyze whether clients were infected with RAASNet [10], an open-source ransomware.

The contributions of this study are as follows.

- We propose a framework for detecting attacks using open-source tools in network and system environments.
- We confirmed that the proposed ransomware countermeasure method was feasible in a real system environment, by examining the characteristics of each open-source tool, selecting the tool's core function, and experimenting.

The rest of this paper is structured as follows: Section II introduces related research. Section III describes our experiments with ransomware detection in three open-source tools, and analyzes the results. Section IV describes a ransomware detection methodology for each open-source tool, and Section V provides the conclusions.

Manuscript received January 19, 2021. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1F1A1061107) and Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008703, The Competency Development Program for Industry Specialist), and a follow-up of the invited journal to the accepted & presented paper entitled "Study on Systematic Ransomware Detection Techniques" of the 23th International Conference on Advanced Communication Technology (ICACT2021).

Sun-Jin Lee is with the Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea (e-mail: 220214013@sungshin.ac.kr)

Hye-Yeon Shim is with the Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea (e-mail: 220214012@sungshin.ac.kr)

Yu-Rim Lee is with the Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea (e-mail: 220214014@sungshin.ac.kr)

Tae-Rim Park is with the Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea (e-mail: 220214011@sungshin.ac.kr)

Il-Gu Lee is with the Department of Future Convergence Technology Engineering, Sungshin Women's University, South Korea (Corresponding Author phone: +82-2-920-7145; e-mail: iglee@sungshin.ac.kr)

II. RELATED WORK

As the number of APT attacks such as ransomware increase exponentially, technologies to prevent and defend against such attacks are being actively studied.

A. Cyber Attack Detection

One study detected attacks using the characteristics of ransomware communication by analyzing the HTTP message sequence and content size of ransomware products [11]. In that paper, network communication results such as HTTP traffic attributes were observed using the representative ransomware families “CryptoWall” and “Locky,” and a detection approach based on Proof of Concept Software Defined Network (SDN) was used. The system achieved a 97% detection rate, showing the feasibility of the approach. However, this approach had a limitation, in that the HTTP response size of the command and control server, which is mainly used for attacks, was not considered. This study differed from ours, in that it detected attacks based on network attribute information.

There have also been studies into systems that defend against attacks by analyzing the characteristics of the ransomware. Representative studies include the detection of Android malware by benchmarking the detection engines of Dr-Droid and Virustotal, open-source malware detection tools [12]. In this study, for malware based on adware, ransomware, scareware, and Short Message Service (SMS), Dr-Droid detected attacks with an accuracy of 79.06% and Virustotal's detection engine detected attacks with an accuracy of 98.7%. The work reported in this paper was similar to this study in that ransomware was detected, but it did not cover artifact information such as files or the time infected with each ransomware.

In some studies, attacks were detected by identifying suspicious behavior in event logs. A representative study related to a score-based anomaly detection technique at endpoints using the Local Outlier Factor (LOF) and Autoencoder [13]. This study used unsupervised learning to detect abnormal behavior as suspicious attacks after learning normal behavior. Suspicious behavior was identified by applying the allowed list operation policy, which resulted in 107 previously undetected suspected processes in LOF, 44 in AutoEncoder-based system behavior, and 24 in network behavior. However, it is difficult to filter all files using this study's proposed operational policy, and there was a limitation in that some false detection occurred because unsupervised learning was used.

B. Attack Detection Using Open-source EDR Tools

Recently, studies have used open-source EDR tools to detect malicious code. A representative study detected exploits using GRR [14]. In this paper, we studied a monitoring system which aimed to track attack threats, and demonstrated that two types of exposures could be detected. In controlled experiments, attacks due to vulnerabilities in FTP servers and Adobe Reader could be detected using GRR, and abnormal behavior of memory and networks was detected using Hunts among the GRR functions. This work demonstrated that cyberattacks are detectable by GRR tools, but there was a limitation, in that it could not determine how to detect cyberattacks using EDR tools other than GRR.

Another study was tested using osquery, flet, and elastic stack solutions, to demonstrate the effectiveness of

open-source EDR systems in real-world environments [15]. In this study, various types of trojan attacks such as “Applejeus: Lazarus” and “jRAT/Adwind” were detected using open-source EDR, demonstrating their applicability in real-world environments. However, there was a limitation, in that it was not possible to confirm whether this approach was effective in all environments, because experiments were not conducted with attacks other than trojan horses.

A previous study evaluated the detectability of attacks using several types of open-source EDR tools [16]. In this study, we analyzed the EDR tools osquery, GRR, and Mozilla InvestiGator (MIG), which can detect attacks using real-time forensic functions. The three real-time forensic tools can collect information by processing vast amounts of data in real-time over a network, without turning off the power to a computer suspected of damage. In this study, three tools were evaluated for process creation, persistence, and network connection. It was possible to monitor the internal operations of the operating system using osquery, find attacked files using GRR and MIG, and detect attacks by searching for strings expected to be malware samples in the actual file content. The authors confirmed that MIG is useful, in that it can handle embedded systems, but was not sufficient to check the Windows registry. The use of osquery demonstrated that a system can be automatically monitored based on a variety of attributes and artifacts, along with the real-time investigation capabilities provided by GRR and MIG. By evaluating the possibility of detecting attacks at the level of process generation, persistence, and network connection among open-source EDR tools, this study attempted to demonstrate the value of the use of EDR tools for ransomware detection in network and system environments.

III. Open-source-based Ransomware Detection

A. Ransomware Detection Frameworks

In this study we used open-source EDR tools to detect when a device was infected by RAASNet, an open-source ransomware. At this time, the payload of ransomware generated by RAASNet is an “override and name” method. This is an attack that copies an existing file, changes the file extension to the .DEMON format, and then removes the existing file to prevent viewing or modifying the encrypted file. The following is a ransomware detection method for each of the representative open-source tools, GRR, osquery, and OSSEC.

1) Google Rapid Response

GRR is a framework for responding to infringement accidents, focusing on remote live forensics developed by Google.

A GRR server communicates with pre-registered clients using messages. A client communicates with a server using the HTTP protocol, and sends a batch of responses via periodic HTTP POST requests. The client generates a message queue based on the client's name, and then processes the responses in the queue. The client may only receive a request corresponding to its own queue, but may transmit a response to all worker queues. Communication between the server and client is encrypted using an RSA public key, signed using the sender's private key [17].

Table I summarizes the core functions provided by GRR.

TABLE I
KEY FEATURES OF GRR

Feature	Description
Cross-platform	Support for cross-platform Linux, OS X and Windows
Memory Analysis	Real-time remote memory analysis using YARA library
Can schedule action queries	Recurring tasks can be scheduled
Registry search function	Search and download capabilities for files and Windows registry
Large-scale host monitoring	Large-scale monitoring using the Hunt function

GRR can run on Linux, OS X (macOS), and Windows environments and enables real-time remote memory analysis using YARA, a tool designed to help malware researchers identify and classify malware samples. It is possible to automate the process using the reservation function for repeatedly performed Flow or Hunt. The user can also search for or download files and the Windows registry. Hundreds of digital forensic artifacts can be collected, and OS-level and native file systems are accessible using SleuthKit (TSK) [18].

In this study, queries about before and after the infection of the victim device systems were performed using File Finder, which can obtain information about files in specific memory spaces during the flow of the GRR server. Using this approach, changes in the file system were checked before and after their infection with ransomware.

2) osquery

osquery is a SQL-based operating system instrumentation, monitoring, and analysis framework developed by Facebook. It has a built-in SQL language and hundreds of tables, so it is effective at responding to incidents. There are two types of osquery: osqueryi and osqueryd. osqueryi is an interactive query console/shell that allows users to explore the state of the operating system through the shell without the need to communicate with a daemon or run as administrator. osqueryd is a host monitoring daemon that can schedule queries and record state changes in an operating system. The daemon aggregates query results over time and analyzes changes in the query's state. The OS event API can be used to log file or directory changes and hardware and network events [19].

Table II summarizes the core functions provided by osquery.

TABLE II
KEY FEATURES OF OSQUERY

Feature	Description
Cross platform	Can be built and used on a variety of OSs
Interactive Query Console	An interactive query console, osqueryi, provides a SQL interface for testing new queries and navigating the operating system.

Can schedule action queries	A high-performance host monitoring daemon osqueryd allows users to schedule queries that run across their infrastructure
Large-scale host monitoring	osqueryd's logging is tech stack-agnostic thanks to its powerful plugin architecture. It can be integrated into the existing internal log aggregation pipeline.

osquery can run in the Windows, OS X, Linux, and FreeBSD environments, and manages the operating system as an interactive query console and relational database. Large-scale environmental monitoring is possible using various plug-in functions.

In this study, all files in the victim's infected memory space were queried using the file query statement of osqueryi. As with GRR, queries were performed before and after ransomware infection to investigate changes in the file system before and after infection.

3) Open-source HIDS SEcurity (OSSEC)

OSSEC is an open-source host-based intrusion detection system that allows administrators to monitor information on agents, syslogs, databases, and agentless devices. OSSEC's agent uses UDP communication, collects information, and forwards information to administrators for analysis. At this time, some information is collected in real time and some is collected periodically [20]. In the absence of an agent, it is possible to monitor firewalls, routers, and Unix systems for agentless devices.

Table III summarizes the core functions provided by OSSEC.

TABLE III
KEY FEATURES OF OSSEC

Feature	Description
Cross platform	Can be built and used on a variety of OSes
File Integrity Check	Detect system changes in real-time to files and Windows registry settings, maintain and manage forensic copies of data
Log monitoring	Monitoring and analysis of multiple log data in real time
Rootkit detection	Process and file-level analysis to detect malicious applications and rootkits
Active response	Real-time response to attacks and changes to systems through firewall policies, integration with third parties such as Content Delivery Networks (CDNs) and support portals, and multiple mechanisms

OSSEC provides comprehensive host-based intrusion across multiple platforms, including Linux, Solaris, Advanced Interactive eXecutive (AIX), Hewlett Packard Unix (HP-UX), Berkeley Software Distribution (BSD), Windows, Mac, and VMware Elastic Sky X (ESX). OSSEC enables real-time change history management of files and Windows registry settings, and specializes in analyzing logs in real time. Various policies can be used to respond to attacks or changes, and users can set alert rules according to their security requirements, enabling the performance of customized actions when an alert occurs.

In these experiments, changes to the ransomware-infected files were detected using the integrity check, which is a function of OSSEC; it sets the space to be checked for integrity in an OSSEC configuration, and detects encryption and file extension changes that exist in the memory space.

B. Experimental Environment

The environment was configured by dividing the server and the client to be attacked. An open-source EDR (GRR, osquery, and OSSEC) system was built on the server, and each client version of the open-source EDR was installed on the client before the files were attacked.

Table IV summarizes the components and versions of the environment used in the experiment.

TABLE IV
COMPONENTS AND VERSIONS USED TO BUILD
THE EXPERIMENTAL ENVIRONMENT

Component	Version
CPU	Intel® Core™ i7 Processor 10700K @ 3.80 GHz
RAM	32.0 GB
Server, Client (agent) OS	Ubuntu 18.04.4 LTS
RAASNet	https://github.com/leony024/RAASNet
GRR	3.4.0-1
osquery	3.1.0
OSSEC	4.3.0

The CPU was Intel's i7 10th generation, and Ubuntu 18.04.4 LTS version was used for the OS in each server and client. The software used was GRR 3.4.0-1, osquery 3.1.0, and OSSEC version 4.3.0.

C. Experimental Process

1) Google Rapid Response

For the experiments, each victim was registered in the GRR Server in advance so that it could be managed as a single agent. Figure 1 shows the attack detection process using GRR.

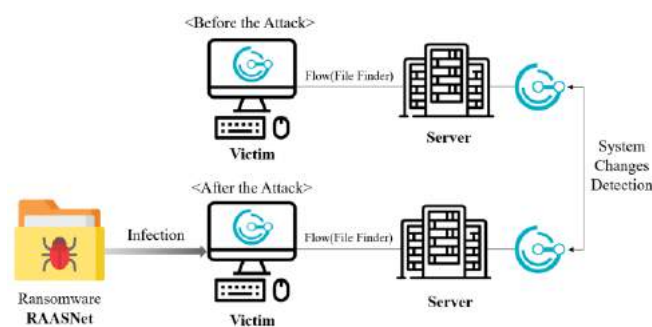


Fig. 1. Configuration of the Google Rapid Response (GRR) experiment environment.

Typical tasks provided by GRR include Flow and Hunt. Flow is a logical collection unit that achieves a given goal in the server and client, and Hunt is a Flow experimental mechanism optimized for a large number of systems. Hunt is used to search for and monitor specific data in a large-scale system, and Flow is used for general experiments. In this experiment, attacks were detected using File Finder during Flow. The attacker attacked a specific folder in the file system of the victim PC registered as a GRR client, and the

GRR Server sent a query to the infected system to compare the query results before and after the attack.

2) osquery

The experiments were carried out by installing osquery on a victim system. Figure 2 shows the attack detection process using osquery.

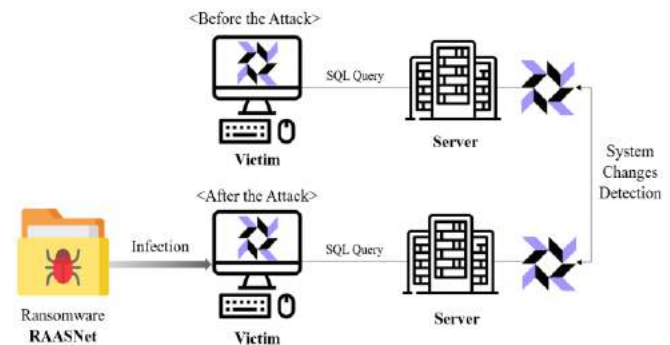


Fig. 2. Configuration of the osquery experiment environment.

In osquery, SQL queries can be written to check and compare data from the operating system. Using a SQL table, it is possible to detect running processes, changes in the hash value of a file, changes in network connections, and other information. osqueryi, an interactive query console, enables incident response, system operation problem diagnosis, and the identification of the causes of performance problems.

In the experiments, we queried osqueryi for file properties in the victim's target folder, and compared the SQL table values before and after infection with RAASNet. We used 'Select * from file WHERE path like "/home/osquery1/target/%"', compared the results using a query, and analyzed the results to check whether the files were infected with ransomware.

3) Open-source HIDS SEcurity (OSSEC)

For the experiments, the victim was registered as an agent in the OSSEC server and system changes were monitored based on the registered ID. Figure 3 shows the attack detection process using OSSEC.

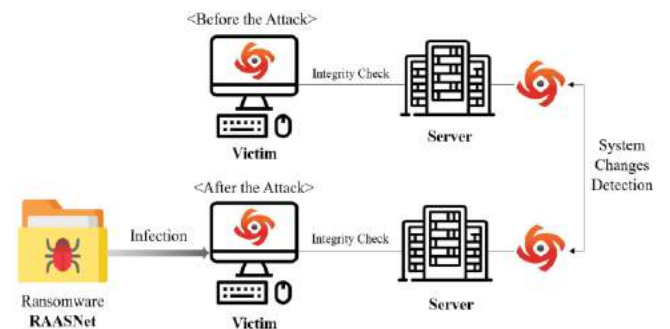


Fig. 3. Configuration of the Open-source HIDS SEcurity (OSSEC) experiment environment.

OSSEC allows users to customize alert rules to meet their security requirements by modifying configuration options. In this experiment, OSSEC's integrity check function was used to detect RAASNet ransomware. The rules for the integrity check were set in the OSSEC configuration, and the monitoring results were checked and compared before and after the attack to determine the ransomware infection status.

D. Experimental Results

To confirm the detection of ransomware, detection was carried out separately before and after RAASNet was executed. For the experiments, 10 MS Office Presentation files, 10 MS Office Word files, and 10 MS Office Excel files were created in a target folder.

1) Google Rapid Response

Figures 4 and 5 show the flow results for one file in the target folder before and after executing RAASNet using GRR File Finder. Checking the experimental result revealed that the atime, mtime, and ctime of the target files in target_folder had changed.

Atime is an abbreviation of last access time, meaning the previous time the file was accessed. The atime value changes when a file is opened or read by a command such as cat or vi. mtime is an abbreviation of last modification time, which means the last time the file was modified. The mtime value is updated when the file contents are added or the file size changes. Ctime is an abbreviation of inode change time, and indicates the last change time of a file. When changing the permission of a file using chmod or changing the owner of a file via chown, the ctime also changes while the permission information is updated in the inode, and ctime is affected when creating or deleting hard links. Unlike atime and mtime, ctime has the characteristic that it cannot be changed with the touch command. When changing data contents, both mtime and ctime are changed.

When the victim's file was encrypted using RAASNet, atime, mtime, and ctime all changed.

Field	Value
Atime	2021-09-28 05:45:09 UTC
Mtime	2021-09-28 04:47:24 UTC
Ctime	2021-09-28 05:46:55 UTC

Fig. 4. Results of Google Rapid Response File Finder Flow before running RAASNet.

Field	Value
Atime	2021-09-28 06:01:55 UTC
Mtime	2021-09-28 06:01:55 UTC
Ctime	2021-09-28 06:01:55 UTC

Fig. 5. Results of Google Rapid Response File Finder Flow after running RAASNet.

2) osquery

Figures 6 and 7 show the query results before and after executing RAASNet using osqueryi, and an analysis of whether the files in the target folder are infected. After executing RAASNet, we used select * from the file with a

path like "/home/osquery1/target/%". When the query was executed, it was confirmed that the name and file size of the ppt1.pptx file, one of the target files, and the atime, mtime, and ctime properties of the file, were changed. As the ransomware RAASNet was executed by targeting the victim's folder, the files in the osquery1 directory were encrypted and detailed properties of the file system changed.

Path	Atime	Mtime	Ctime
/home/osquery1/target/pp1.pptx	1633356399	1631710044	1633356399

Fig. 6. Results of osqueryi before RAASNet execution.

Path	Atime	Mtime	Ctime
/home/osquery1/target/pp1.pptx.DEMON	1633357507	1633357507	1633357507

Fig. 7. Results of osqueryi after RAASNet execution.

3) Open-source HIDS SEcURITY (OSSEC)

OSSEC provides real-time monitoring capabilities. In these experiments, modifying the ossec.conf file changed the detection time in real time, and the detection range was changed to the target folder range. Using a dump of the database and integrity checking on the administrator screen, the status of files in the target folder could be checked, as shown in Figure 8. After RAASNet was infected and the monitoring results were observed through database dumping, it was confirmed that a file was created, as shown in Figure 9, and that the file name and file hash value were changed.

Path	MD5	SHA1
/home/cselab/target_folder/pp1.pptx	md5 81935370e1459b78315d3ddaa83ee0	sha1 31ab584d2b78b55783021eb0029968c37932114

Fig. 8. Hash values of the files in the target_folder directory before checking RAASNet execution using Open-source HIDS SEcURITY (OSSEC)

```

/home/cselab/target_folder/xlsx.DEMON md5 9569240d3a0e3311bd48b2d9c6e9
sha1 3417abf1be35b223b79bfb7a86caaac4012fe
/home/cselab/target_folder/xlsx1.DEMON md5 85d25347c065ae128478958458832f0f0
sha1 19ae73d9d0161dffa59ef4f917af6b3c380cbe7
/home/cselab/target_folder/word7.docx.DEMON md5 9205c45da56427e3dded9fc3517c704f
sha1 f9a90440560855268da1d6508ed363179a57925
/home/cselab/target_folder/xlsx6.xlsx.DEMON md5 181b22074929d8554c507e3c9f2821
sha1 5497d3cb3c082010cf6b83379ac8630012db8bc
/home/cselab/target_folder/xlsx10.xlsx.DEMON md5 1353f7c8af38f226e9dab098327e929e
sha1 d84b21ac40774a40385435c190d0b1310a09c8b
/home/cselab/target_folder/ppt10.pptx.DEMON md5 1c35a277e764024149c91aaf3e99b02
sha1 c5a1aee1d0d5612427ad655c44b3b03118097019
/home/cselab/target_folder/word10.docx.DEMON md5 1947f84ea1e485f27406692667c6a33
sha1 92448b9824cc05968d69b32b7cf742e20c27b15
/home/cselab/target_folder/xlsx7.xlsx.DEMON md5 37384e89786fac2a702a9dc7660083ea
sha1 f28748db289398451a7a4087870b193d19b730
/home/cselab/target_folder/word3.docx.DEMON md5 724bb80cc5e31308e04cc8e254a1d73
sha1 9596a57010c8f352a8161dc879a986dc3c3155
/home/cselab/target_folder/ppt1.pptx.DEMON md5 6d2cdd3fc9832f99d7670a18ca5251
sha1 ec6312c59a2e49dee8d9d48429097c2bfc3ed
/home/cselab/target_folder/xlsx4.xlsx.DEMON md5 6625883861eaf2894ca6a01cf9a0d01
sha1 34004996c99ab1a84718eef6d4fa766088c40
/home/cselab/target_folder/word2.docx.DEMON md5 e144ad3c897ee1d376586f31f228f79e
sha1 5680385ac368d946a7f9e4a92b0a0de8502392
    
```

Fig. 9. Hash values of the files in target_folder directory after checking RAASNet execution using Open-source HIDS SECurity (OSSEC).

IV. ANALYSIS OF EXPERIMENTAL RESULTS

In this study, we confirmed the ability of open-source EDR to detect when the target PC was infected with ransomware. For GRR, Flow's File Finder was used. Attacks could be detected by the changes to the target file name and the file's atime, mtime, and ctime properties. In the case of osquery, the file system of the target PC could be checked using the query statement provided by itself, and attacks could be detected by changes to the target file name and the atime, mtime, and ctime file properties. In the case of OSSEC, the integrity check function was used among the constant monitoring functions provided, and attacks could be detected by changing the file name and hash value. Table V shows the functions, methods, and results of ransomware attack detection by the open-source tools used in this study.

By utilizing the key features of each open-source EDR tool, it was possible to quickly detect various cyberattacks, including ransomware. As shown in Table V, GRR can check condition changes such as the name of the attack target file and the atime, mtime, and ctime properties using the file finder among the filesystems in the flow it provides. In the

case of osquery, it was possible to recognize changes in atime, mtime, and ctime in the file system by querying a database that is updated in real-time for information about the target file. OSSEC uses its own real-time integrity monitoring function to check changes in the attack target file by changing the hash value, such as md5 or sha1, of each attack target file. All three open-source tools detect attacks by changing the environment of the ransomware-infected system, and when a file's hash value and file attribute values change suddenly, it is judged as abnormal behavior and can prompt a real-time response.

V. CONCLUSIONS

As cyberattacks increase and spread, it is essential to quickly detect and take action when attacks occur, to minimize damage to the system. Therefore, it is necessary to introduce tools and information collection systems that can detect attacks in real-time.

In this study we analyzed the unique characteristics and functions of each EDR tool. The target PC's file system was attacked using RAASNet, an open-source ransomware. The possibility of attack detection in open-source EDR was confirmed using GRR, osquery, and OSSEC.

The results of these experiments show that GRR detects attacks through a file finder that can check for file changes during flow, osquery detects attacks through a query statement that detects system changes, and OSSEC detects attacks through a constant monitoring system.

This paper demonstrates the potential of detecting attacks and warning users in the form of logs or alerts. In a follow-up study, we plan to check the detection efficiency of each open-source EDR tool in a large-scale environment, and determine the scope of the response.

TABLE V
ANALYSIS OF RANSOMWARE DETECTION METHODS USING OPEN-SOURCE TOOLS

Open-source software	Detection Function	Detection Section	Detection Method	Detection Result
GRR		File Finder	Select * from file where path like "/target directory/%";	Filename, Atime, Mtime, Ctime
osquery	File System Changes	SQL Query		
OSSEC		Integrity Check	Database Dumping	Hash(md5, sha1)

REFERENCES

1. S. Aurangzeb, M. Aleem, M. Iqbal, and A. Islam, "Ransomware: A Survey and Trends," *Journal of Information Assurance and Security*, Vol. 6, No. 12, pp. 48–58, 2017.
2. O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020.
3. P. H. Meland, Y. F. F. Bayoumy, and G. Sindre, "The Ransomware-as-a-Service economy within the darknet," *Computers & Security*, vol. 92, 2020.
4. "Annual number of ransomware attacks worldwide from 2016 to 2020" Statista. [Online]. Available: <https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/>. [Accessed: 12-Nov-2021].
5. "Cyber Attack Trends – 2021 Mid Year Report" Check Point Research. [Online]. Available: <https://research.checkpoint.com/2021/check-point-software-mid-year-attack-trends-report-reveals-a-29-increase-in-cyberattacks-against-organizations-globally/>. [Accessed: 12-Nov-2021].
6. W. U. Hassan, A. Bates, and D. Marino, "Tactical Provenance Analysis for Endpoint Detection and Response Systems," *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1172–1189, 2020.
7. "GRR Flows," GRR Flows - GRR documentation. [Online]. Available: <https://grr-doc.readthedocs.io/en/v3.2.1/investigating-with-grr/flows/what-are-flows.html>. [Accessed: 10-Nov-2021].
8. "Anomaly detection with osquery," osquery. [Online]. Available: <https://osquery.readthedocs.io/en/stable/deployment/anomaly-detection/>. [Accessed: 10-Nov-2021].
9. "Syscheck," OSSEC. [Online]. Available: <https://www.ossec.net/docs/docs/manual/syscheck/index.html>. [Accessed: 10-Nov-2021].
10. leonv024, "leonv024/RAASNet," GitHub. [Online]. Available: <https://github.com/leonv024/RAASNet>. [Accessed: 10-Nov-2021].
11. K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Computers & Electrical Engineering*, vol. 66, pp. 353–368, 2018.
12. Samara, M., & El-Alfy, E.-S. M., "Benchmarking Open-Source Android Malware Detection Tools," *2019 2nd IEEE Middle East and North Africa COMMUNICATIONS Conference (MENACOMM)*. 2019. doi:10.1109/menacomm46666.2019.8988
13. Kim, Sujeong & Hwang, Chanwoong & Lee, Taejin, "Anomaly Based Unknown Intrusion Detection in Endpoint Environments," *Electronics*. 9. 1022. 2020. 10.3390/electronics9061022.
14. H. Rasheed, A. Hadi, and M. Khader, "Threat Hunting Using GRR Rapid Response," *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pp. 155–160, 2017.
15. Christopher Hurlless, "Exploring Osquery, Fleet, and Elastic Stack as an Open-source solution to Endpoint Detection and Response," *SANS Institute Reading Room site*, 2019. Available: <https://www.sans.org/reading-room/whitepapers/detection/paper/39165>. [Accessed: 11-Nov-2021].
16. Kieseberg, Peter & Neuner, Sebastian & Schrittwieser, Sebastian & Schmiedecker, Martin & Weippl, Edgar. (2018). Real-Time Forensics Through Endpoint Visibility. 10.1007/978-3-319-73697-6_2.
17. M.I. Cohen, D. Bilby, G. Caronni, "Distributed forensics and incident response in the enterprise," *Digital Investigation*, Volume 8, Supplement, Pages S101-S110, ISSN 1742-2876, 2011. <https://doi.org/10.1016/j.diin.2011.05.01>
18. "What is GRR?" GRR. [Online]. Available: <https://grr-doc.readthedocs.io/en/latest/what-is-grr.html> [Accessed: 7-Dec-2021]
19. "Welcome to osquery" osquery. [Online]. Available: <https://osquery.readthedocs.io/en/stable/>. [Accessed: 11-Nov-2021].
20. "OSSEC Architecture" OSSEC. [Online]. Available: <https://www.ossec.net/docs/docs/manual/ossec-architecture.html>. [Accessed: 11-Nov-2021].



Sun-Jin Lee was born in Korea in 2000. She is a student of the Integrated B.S./M.S. course in the Department of Convergence Security Engineering in Sungshin Women's University, Seoul, Korea. Her current research interests are in the area of deep learning, Internet of Things, malware detection, voice security, image security, and video security.



Hye-Yeon Shim was born in Korea in 2000. She is a student of the Integrated B.S./M.S. course in the Department of Convergence Security Engineering in Sungshin Women's University, Seoul, Korea. Her current research interests are in the area of artificial intelligence, deep learning, malware detection, and programming.



Yu-Rim Lee was born in Korea in 1999. She is a student of the Integrated B.S./M.S. course in the Department of Convergence Security Engineering in Sungshin Women's University, Seoul, Korea. Her current research interests are in the area of artificial intelligence, threat defense, malware detection, and Internet of Things.



Tae-Rim Park was born in Korea in 1999. She is a student of the Integrated B.S./M.S. course in the Department of Convergence Security Engineering in Sungshin Women's University, Seoul, Korea. Her current interests are in the areas of artificial intelligence, network security, malware detection, cloud computing, and Internet of Things.



Il-Gu Lee was born in Korea in 1978. He received his PhD degree in the Graduate School of Information Security in Computer Science & Engineering Department from KAIST at 2016. He is a professor at the Department of Convergence Security Engineering, Sungshin Women's University, Seoul, Korea. His current research interests are in the area of wireless/mobile networks with an emphasis on information security, networks, and wireless systems.