ORIGINAL ARTICLE

ETRI Journal WILEY

# Fileless cyberattacks: Analysis and classification

GyungMin Lee[1] | ShinWoo Shim[2] | ByoungMo Cho[2] | TaeKyu Kim[2] |
Kyounggon Kim[1,3] (iD)

[1]School of Cybersecurity, Korea University, Seoul, Rep. of Korea

[2]Intelligent SW Research Center, LIG Nex1, Seoul, Rep. of Korea

[3]Department of Forensic Sciences, Naif Arab University for Security Sciences, Riyadh, Kingdom of Saudi Arabia

**Correspondence**
Kyounggon Kim, School of Cybersecurity, Korea University, Seoul, Rep. of Korea.
Email: anesra@korea.ac.kr

**Funding information**
This research was supported by LIG Nex1.

**Abstract**

With cyberattack techniques on the rise, there have been increasing developments in the detection techniques that defend against such attacks. However, cyber attackers are now developing fileless malware to bypass existing detection techniques. To combat this trend, security vendors are publishing analysis reports to help manage and better understand fileless malware. However, only fragmentary analysis reports for specific fileless cyberattacks exist, and there have been no comprehensive analyses on the variety of fileless cyberattacks that can be encountered. In this study, we analyze 10 selected cyberattacks that have occurred over the past five years in which fileless techniques were utilized. We also propose a methodology for classification based on the attack techniques and characteristics used in fileless cyberattacks. Finally, we describe how the response time can be improved during a fileless attack using our quick and effective classification technique.

**KEYWORDS**

classification, cyber security, cyberattack, fileless malware

## 1 | INTRODUCTION

The infrastructure in Estonia was brought down by a cyberattack from a suspected Russian state-sponsored hacker group in April 2007 [1]. Nations with strong cyber superstructures, such as China, North Korea, and Russia, have reinforced their cyberattack capabilities [2]. The sources of such cyberattacks are now moving from individuals to organized hackers supported by governments, and their cyberattacks are progressing into more complex and advanced initiatives, which were previously unlikely to develop from individual hackers [3].

Before cyberattacks evolved into more intricate attacks, simple security solutions such as virus protectors could be used to block them; however, attackers began to utilize various attack strategies to improve their effectiveness. Since 2014, fileless cyberattacks have been continuously on the rise owing to the fact that they cannot be detected by vaccines and can circumvent even the best efforts of security analysts. However, despite the analysis of individual fileless malware conducted by security companies, studies on fileless cyberattacks in their entirety remain insufficient. Therefore, in this paper, such attacks are analyzed, summarized, and classified based on cases that have emerged since the mid-2010s.

The following are the three main contributions of this study.

- An analysis of the detailed attack techniques of 10 types of fileless cyberattacks.
- A mapping of the cyber kill chain attack stage with each fileless cyberattack technique.
- A suggested classification methodology for the 10 fileless cyberattacks.

The rest of this paper is organized as follows. In Section 2, we review related studies to the current research. In Section 3, we present our methodology for classifying fileless cyberattacks and describe a detailed procedure and analysis of the results for each methodology. In Section 4, we discuss fileless cyberattacks compared with traditional malware. In Section 5, some concluding remarks are provided along with an outline of the potential implications of this study and suggestions for future studies in this field.

## 2 | LITERATURE REVIEW

Various studies on fileless cyberattacks have been conducted. For example, to identify fileless cyberattacks against Linux-based Internet-of-Things machines, Dang and others designed a software- and hardware-based honey pot and collected data on malicious code for approximately one year [4]. They confirmed that among the malicious code collected, 10% were fileless cyberattacks, which were then classified into eight groups using the characteristics of the corresponding attacks. They analyzed the attacks by focusing on their characteristics and methods of defense.

Sanjay and others classified fileless cyberattacks into two categories: memory- and script-based attacks [5]. In addition, to overcome their defense mechanism, they classified the strategies used by fileless cyberattacks into four categories. They also listed the mechanisms that can be used for detecting or even defending against fileless cyberattacks. The representative technique of fileless cyberattacks introduced herein involves a document containing a malicious code, and the detection and defense technique involves the use of an analysis program such as Yara or an operating system function such as Microsoft Enhanced Mitigation Experience.

Rivera and inocencio comprehensively analyzed Poweliks, a representative fileless cyberattack. They also analyzed various relevant attacks and tools such as Phasebot, Gootkit, and Emotet and examined the strategies used by each attack [6]. Furthermore, by analyzing various attacks, they suggested four strategies for defending against fileless cyberattacks.

Kumar and sudhakar investigated seven representative fileless cyberattack samples and classified them based on persistent techniques [7]. They classified attack samples based on an investigation of the attack vectors for each cyberattack. The categories were determined as follows: memory resident malware that resides in the system memory, window registry malware that hides in the system registry, and rootkit fileless malware. They extended their study and suggested a framework for countermeasures that can be used when fileless cyberattacks are executed on a system.

In addition, O'Murchu and others analyzed in detail how Poweliks changed in 2015 [8]. The researchers analyzed and compared the specific functions from Poweliks 1.0, the earliest version, to Poweliks 1.7, a relatively recent version, and checked the fileless cyberattack specifics of Poweliks. According to a Symantec analysis, Poweliks uses mechanisms that protect the registry keys and strategies developed to obtain CLSID information and authority elevation, thereby making it difficult for users to identify infections.

Lee and others analyzed Poweliks and Kovter, which can be described as representative fileless malware [9,10]. Through their analysis, they described how fileless malware conceals its activity using the registry and memory and suggested a method for detecting fileless malware based on such use.

## 3 | METHODOLOGY

Our detailed methodology is illustrated in Figure 1. We collected public fileless cyberattack samples at public sites such
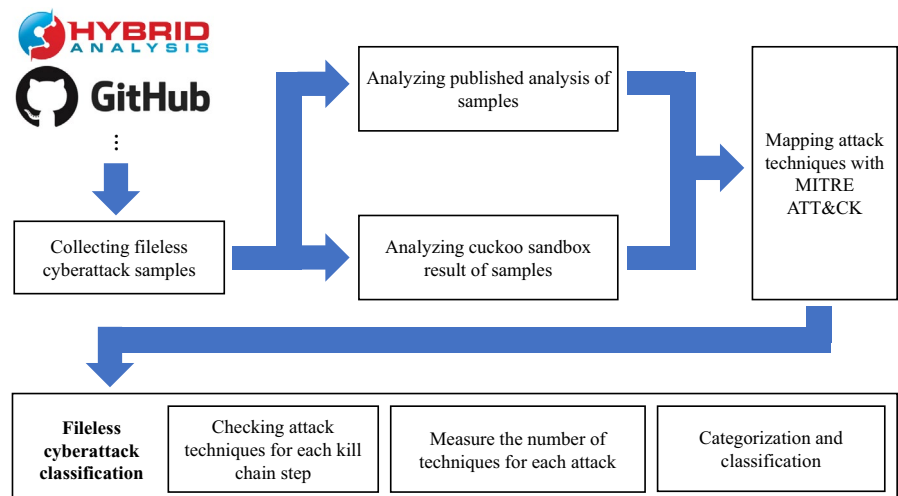


**FIGURE 1** Methodology for classification of fileless cyberattacks

as Hybrid Analysis and GitHub and then analyzed the techniques used by fileless malware through open source intelligence. We then used Cuckoo Sandbox to extract the results of a fileless cyberattack analysis. Next, we used the collected information to map the fileless malware attack techniques using the ATT&CK kill chain published by MITRE and analyzed the attack techniques. Finally, we scored and classified the fileless cyberattacks.

## 3.1 | Collecting fileless cyberattack cases

As presented in Table 1, in this study, we investigated and collected a variety of known fileless cyberattacks as samples. These samples were collected using either the cyberattack dataset published by GitHub or the dataset published by Hybrid Analysis, which is a German dataset of malicious code.

When users upload malicious code samples, the aforementioned website analyzes the code that uses various anti-virus products. On the Hybrid Analysis website, users can infer the identity of malicious code using the analysis results and a self-registered tag. Therefore, we searched Hybrid Analysis for malicious code based on the tags. From the search results, we selected appropriate malicious code according to the output screens of the analysis results, malicious features, hash values of the malicious code, and published data.

## 3.2 | Analysis of fileless cyberattack malware

We analyzed ten fileless cyberattacks to identify the specific techniques used by each, and in the following sections, we provide an in-depth analysis for each type of attack.

### 3.2.1 | Poweliks

In 2014, Poweliks was the very first fileless malicious code to be detected. This code spreads through malicious host files attached to emails. According to the security company, G Data, malicious files have been delivered through email messages impersonating the international freight transportation company, UPS [11]. These spurious UPS files infiltrate PCs using the macro vulnerability of MS Word. Poweliks is a fileless attack because its information is stored in the registry to avoid detection by users and to ensure permanent infection.

### 3.2.2 | Rozena

Rozena, which was discovered in 2015, deceives users by disguising itself as a normal MS Word file. According to a 2018 report by a German security company [12], Rozena runs through several PowerShell scripts and shows fileless characteristics in that it inserts the malicious code into the memory through scripts. Rozena uses PowerShell to remain in the memory and communicate with the attacker's PC.

### 3.2.3 | Duqu 2.0

Duqu 2.0 was developed as a part of the advanced persistent threat attack against Kaspersky in 2015 [13]. Duqu 2.0 is the updated version of the 2011 Duqu 1.0 attack and is known to be related to Stuxnet. It uses an injection technique to reside in memory for only a short time. However, this allows Duqu 2.0 to operate within the system for an extended period. No files are stored on a host infected by Duqu 2.0, and the files generated during the attack are deleted by the main module.

**TABLE 1** Fileless cyberattack samples used in the experiment

| No | Fileless cyberattack | Year | Size | SHA256 |
|---|---|---|---|---|
| 1 | Poweliks | 2014 | 70 KiB | - |
| 2 | Rozena | 2015 | 593 KiB | c23d6700e93903d05079ca1ea4c1e36151cdba4c5518750dc604829c0d7b80a7 |
| 3 | Duqu 2.0 | 2015 | 254 KiB | 52fe506928b0262f10de31e783af8540b6a0b232b15749d647847488acd0e17a |
| 4 | Kovter | 2016 | 331 KiB | - |
| 5 | Petya | 2017 | 354 KiB | 027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745 |
| 6 | Sorebrect | 2017 | 807 KiB | 4142ff4667f5b9986888bdcb2a727db6a767f78fe1d5d4ae3346365a1d70eb76 |
| 7 | WannaCry | 2017 | 3432 KiB | ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa |
| 8 | Magniber | 2017 | 48 KiB | c21887eaa1e31b9220d0807d3a7d0f30421ab6f80cfc1c556d534587dd9e6343 |
| 9 | Emotet | 2017 | 145 KiB | 70903a9ef495edd8de01a61f8e9862a037b0dee327d7f92f93ef69e33e461764 |
| 10 | GandCrab | 2018 | 125 KiB | 643f8043c0b0f89cedbfc3177ab7cfe99a8e2c7fe16691f3d54fb18bc14b8f45 |

### 3.2.4 | Kovter

Kovter was discovered in 2016 and can be described as malware with several functions, similar to Poweliks. According to an analysis conducted by the security company Check Point [14], Kovter also spreads through malicious files attached to emails; however, this malware runs malicious files through JavaScript in the attached file. Specifically, it runs JavaScript through a Mshta.exe file, refers to information outside of the registry, and saves the malicious information to the registry.

### 3.2.5 | Petya and NotPetya

The Cybersecurity and Infrastructure Security Agency, a US government agency, published an analysis on Petya and NotPetya in 2017 [15]. NotPetya, a variant of Petya, is a malware with some added functions beyond those of Petya. NotPetya spreads through malicious files attached to emails or malformed programs. It has a feature in which the malware is run without the need to install a special program on another host, for example, through the EternalBlue exploit, which uses the SMB vulnerability.

### 3.2.6 | Sorebrect

Identified in 2017, a Sorebrect fileless attack can be described as ransomware that does not store information in the registry. With remote access installed, Sorebrect obtains account information using a brute force attack and remotely runs malware on the victim hosts using PsExec. Sorebrect uses a code injection technique to implant its malicious function into svchost.exe. Sorebrect also deletes various event-log related artifacts such as compat, shimmachi, and prefetch. This code is inserted through a code injection-enacted file encryption after the original binary files have been deleted.

### 3.2.7 | WannaCry

According to a 2017 analysis report published by FireEye, WannaCry ransomware exploits EternalBlue, an SMB vulnerability disclosed by the National Security Agency [16]. This malware has characteristics of both ransomware and worms and therefore encrypts files on the victim hosts while simultaneously transferring the files to other hosts connected to the network. There are no trace files left on the victim host because WannaCry inserts a shellcode into the SMB payload and sends it through the network packet.

### 3.2.8 | Magniber

Magniber, which can be described as ransomware that has mainly existed domestically in the Rep. of Korea since 2017, has mostly targeted hosts using the Hangeul version of the Windows operating system. According to an analysis report by the vaccine company MalwarebytesLab, Magniber infects the user host by taking advantage of the VBScript remote code execution vulnerability [17]. To hide traces of malicious actions, Magniber deletes the data restored by the Windows operating system, known as a shadow copy, and prevents the restoration of the host.

### 3.2.9 | Emotet

Emotet first appeared in 2014 and targeted German banks. It reappeared in 2017 with more advanced functions. AhnLab, a security and vaccine company, published an analysis report on this malware [18]. According to this report, the first distribution of Emotet was an email containing an attached malicious file. This attachment looks like an MS Word or PDF file, and it initiates an attack when a victim enables the macros in that malicious file. Similar to other malware, Emotet also runs an obfuscated script mainly using PowerShell and acts as a dropper that downloads other malicious files from the attacker's server.

### 3.2.10 | Gandcrab

Gandcrab is a type of ransomware that has a range of activities that includes targeting browser vulnerabilities, disguising itself as normal software, or causing infections through malicious documents attached to emails. According to an analysis report published by AhnLab, Gandcrab initially runs an attack tool called Magnitude through a distribution script inserted on a website [19]. This attack tool runs certain dll files on the victim's PC and inserts a ransom code into svchost.exe, which is a commonly used process for additional malware execution.

## 3.3 | Analysis of detailed techniques in fileless cyberattacks

We utilize Cuckoo Sandbox to analyze the techniques used in fileless cyberattacks in detail. This section describes the Cuckoo Sandbox analysis process for 10 fileless cyberattacks. We then map each technique using the MITRE ATT&CK framework.

### 3.3.1 | Cuckoo Sandbox analysis

Cuckoo Sandbox is an open malware analysis system that extracts and provides malware information based on the actual operation of the malware in a virtual environment. The static analysis information, which is basic malware information, refers to portable executable (PE) information and resource information on the process. This information can be checked through a Cuckoo Sandbox static analysis report. One of the critical aspects of any malware is information on the actions taken by the malware against a PC. Cuckoo Sandbox displays a signature as units of information for each malicious action. Through such a signature, it is possible to check the contents and risk-based information of malicious code. As an additional feature, Cuckoo Sandbox can reveal the characteristics of the dropper, which generates additional files. Using the identified characteristics, information on the dropped files and screenshots of the malware running in a virtual environment can be checked. After the dropper files have been identified, the signature tags identified by the Cuckoo Sandbox report are analyzed to check whether the signatures are techniques that the actual malware has also applied.

The results of the Cuckoo Sandbox analysis for our 10 fileless cyberattack samples are detailed in Table 2. The table also displays the average characteristics that can be universally identified after three analysis runs for each type of malware using Cuckoo Sandbox. The values set by Cuckoo Sandbox are based on the statistics of the number of dropped files and buffers, the number of network hosts that a PC with malware installed attempts to connect to, the number of processes that the malware runs, the PE section as dynamic information, the number of imports, and the number of resources.

In Table 2, *Score* represents the number calculated by Cuckoo Sandbox and is an indicator of the risk of the malware. In addition, *Number of signatures* refers to the number of signatures for each malicious code. This information contains signatures that are suspected to be malicious, and the signatures that normal programs can hold. The higher the number of signatures that are suspected of malicious behavior among the *Number of signature* indicators, the higher the score.

The *Number of file drops* refers to the number of files that malicious code installs or downloads on the user's PC. The *Number of net hosts* indicator refers to the number of other hosts on an external network to which the malicious code is connected. For all fileless malware, it was found that one host is connected to the outside by default, although it was also confirmed that Internet Explorer was connected to msn.com. In addition, *Number of processes* refers to the number of processes generated by the malicious code, and in the case of malicious code such as Poweliks, it was confirmed that processes continuously regenerate themselves. The *Number of API calls* indicator represents the overall number of API calls generated by malicious code. This is a representation of the APIs that are called by all created processes. In addition, *Number of sections*, *Number of imports*, and *Number of resources* refer to information that can be checked in the PE structure. For example, the number of sections in the PE structure of each malicious code, number of programs identified in the import section, and number of resources included.

To obtain more accurate information, we cross-referenced the malware techniques with those presented by a published report and analysis. Examples of the techniques used, as identified in the report, are listed in Table 3. We also checked the techniques applied by certain types of malware, by conducting static and dynamic malware analyses.

Table 3 lists the attack techniques used by Poweliks and Kovter, both of which are representative fileless cyberattacks. We confirmed whether the analysis of information provided by Cuckoo Sandbox is similar to that based on the published analysis reports from Cuckoo Sandbox and the actual analysis. Based on this, we found that the technique used by each

**TABLE 2** Summary of the Cuckoo Sandbox results for 10 sample fileless cyberattacks

| Cyber attack | Score | Number of sig. | # of file drops | # of net hosts | # of proc. | # of API calls | # of sections | # of imports | # of resources |
|---|---|---|---|---|---|---|---|---|---|
| Poweliks | 15.2 | 31 | 2 | 2 | 49 | 5823 | 6 | 4 | 1 |
| Rozena | 11.6 | 25 | 11 | 2 | 6 | 23 | 16 | 3 | 26 |
| Duqu 2.0 | 3.2 | 6 | 0 | 1 | 1 | 67 | 4 | 4 | 0 |
| Kovter | 20.4 | 42 | 5 | 158 | 5 | 4990 | 8 | 10 | 26 |
| Petya | 3.8 | 8 | 0 | 1 | 1 | 78 | 5 | 13 | 4 |
| Sorebrect | 11.2 | 22 | 1 | 5 | 2 | 232 | 5 | 0 | 10 |
| WannaCry | 24.8 | 49 | 1113 | 14 | 30 | 24 511 | 4 | 4 | 3 |
| Magniber | 7.6 | 17 | 617 | 1 | 9 | 81 | 4 | 3 | 0 |
| Emotet | 9.6 | 18 | 0 | 5 | 3 | 268 369 | 4 | 5 | 11 |
| GandCrab | 4.4 | 10 | 0 | 1 | 3 | 353 | 5 | 5 | 9 |

**TABLE 3** Techniques used in two sample cyberattacks

| Fileless cyberattack | Techniques |
| --- | --- |
| Poweliks | MS Office macro vulnerability |
| | Injection of malicious scripts into the registry |
| | Execution of registry value using Rundll32.exe |
| | Execution of registry value encoded using Jscript. Encode |
| | Use of PowerShell scripts encoded with Base64 |
| | Verification of the registry key and path of executed files |
| | DLL execution through PowerShell scripts (injection using dllhost.exe) |
| | Deletion of files after every operation |
| | Resides in Dllhost.exe |
| | Sending a user's system information to the C&C server through TCP communication |
| Kovter | Social engineering techniques using email attachments |
| | Injection of malicious script into the registry |
| | Execution of registry values using Mshta.exe |
| | Execution of registry values encoded using Jscript.Encode |
| | Use of PowerShell scripts encoded with Base64 |
| | Injection of code through PowerShell scripts |
| | Deletion of files after every operation |
| | Resides in Regsvr32.exe |
| | Sending a user's system information to the C&C server through TCP communication |

malware provided by the Cuckoo Sandbox report does not entirely agree with the actual analysis conducted in the present study; however, most of the techniques are described in the signature information. We therefore conducted an experiment based on the results of Cuckoo Sandbox for each type of malware.

### 3.3.2 | MITRE ATT&CK mapping

MITRE ATT&CK is a framework that organizes the techniques used by malware and cyberattacks into 12 stages with the technique of each stage defined in detail. We investigated the steps of the various malicious techniques used in fileless cyberattacks. Table 4 lists the results of mapping 10 fileless cyberattacks using MITRE ATT&CK based on the published analysis report, the analysis results of the present study, and the results of Cuckoo Sandbox.

Each row in Table 4 represents one stage of a cyberattack as classified by MITRE ATT&CK, and each column

represents the 10 types of fileless cyberattacks we investigated. For example, the information in the "Poweliks" column of the "Defense evasion" row: "modify registry," "process injection," "software packing," "file deletion," and "obfuscated files or information," shows the attack techniques Poweliks uses for the purpose of evading a defense. Typically, the modify registry technique is a technique in which malicious code changes the user's registry to store data or perform a malicious action. In addition, a process injection refers to a technique in which malicious code is inserted into a normal process and is executed to conduct a malicious action without being detected by the user. We verified our findings in Table 4 by cross-referencing them with the results presented in other published reports.

To check the list of techniques at each stage used by each type of malware, we analyzed all techniques, as depicted in Figure 2. In this figure, each layer represents a classification of the stages of a malicious code attack in MITRE ATT&CK. The order of the layers corresponds to the flow of an attack, and we analyze the techniques used in each of these stages for the 10 fileless cyberattacks. Each red line in this figure connects a technique used by malicious code with the previous technique it has used. For example, it was found that the Magniber and Gandcrab cyberattacks used the *Drop by compromise* technique, which infects users through compromised websites, during the initial access stage. Next, we identified that the Magniber and Gandcrab cyberattacks use the following techniques during the execution step: a *Scheduled task* to automatically run the program using the Windows Scheduler, an *Exploit technique* using a direct system vulnerability, and a *WMI technique* for code execution.

At each stage, the shade of the blue plane indicates the type of technique used. In addition, the color of the circle representing the attack technique indicates the number of malicious code types that use it. For instance, if the number of malicious code types is five or more, the color of the circle is dark brown; if it is three or four, it is orange, and if it is two or less, it is an apricot color.

A defense evasion is applied in most types of malicious code, and these techniques enable malicious code to avoid being noticed by users or experts. Because of these two characteristics, we can conclude that script-based attacks are the most frequent attacks that employ various programs such as PowerShell and WMI to operate without leaving any trace on the user's PC. In addition, it can be seen that various defense evasion based techniques are used to conduct attacks and remove traces of the user to avoid being detected at a later stage.

### 3.4 | Classification of fileless cyberattacks

This study classifies fileless cyberattacks based on the results derived from Cuckoo Sandbox and techniques found

**TABLE 4** Mapping of fileless cyberattack techniques using MITRE ATT&CK

| MITRI ATT&CK | Poweliks | Rozenna | Duqu 2.0 | Kovter | Petya |
|---|---|---|---|---|---|
| Initial Access | Spearphishing attachment | Spearphishing attachment | Spearphishing attachment | Spearphishing attachment | Spearphishing attachment, Supply chain compromise |
| Execution | Script, Rundll32, Scripting, PowerShell | User execution, Scripting, PowerShell | Signed binary proxy execution | Mshta, Scripting, PowerShell, Regsvr32 | Scripting, Mshta, Service execution, Windows management instrumentation, Rundll32, Scheduled task |
| Persistence | Registry run keys | - | Scheduled task | Registry run keys | - |
| Privilege Escalation | Process injection | Process injection | Exploitation for privilege escalation, Access token manipulation | - | - |
| Defense Evasion | Modify registry, Process injection, Software packing, File deletion, Obfuscated files or information | Deobfuscate/Decoded files or information, File deletion, Obfuscated files or information | Disabling security tools | Modify registry, File deletion, Obfuscated files or information, Deobfuscate/, Decoded files or information | Mshta, Indicator removal on host |
| Credential Access | - | - | Credential dumping | - | Credential dumping |
| Discovery | - | - | Process discovery, Account discovery, Network share discovery, Network service scanning | - | File and directory discovery |
| Lateral Movement | - | - | Pass the hash, Windows admin shares | - | Windows admin shares, Exploitation of remote services |
| Collection | Data from local system | - | Data from local system | Data from local system | - |
| Command and Control | Commonly used port | Commonly used port | Commonly used port | Commonly used port | - |
| Exfiltration | Automated exfiltration, Exfiltration over alternative protocol | - | Data encrypted | Automated exfiltration | - |
| Impact | - | - | - | - | Disk structure wipe, Data encrypted for impact |

from analysis reports. In this study, based on the malware analysis results of Cuckoo Sandbox, our intent is to lay the foundation for the classification of cyberattack files using the MITRE ATT&CK signature information. Cuckoo Sandbox provides mapping information for each signature discovered by MITRE ATT&CK. However, MITRE ATT&CK does not provide information for all types of signatures, and when compared with published reports, there are differences between the findings of MITRE ATT&CK and those of other in-depth analyses.

The signature information of cyberattacks can be extracted from the JSON-type report generated by Cuckoo Sandbox. For example, MITRE ATT&CK information can be obtained from the TTP information presented in each signature. The collected information is saved separately in the form of a CSV file for graph generation for later classification and saves information of the 12 MITRE ATT&CK stages for each malware type. We classify fileless cyberattacks into three types: Attack, Evasion, and Collection attacks. Of the 12 MITRE ATT&CK stages, the Privilege Escalation and Impact stages represent a collection of techniques that malware generally use for destroying information and systems as well as for interfering with the normal actions of a PC. Based on our classification, malware is classified as a "fileless attack" if a substantial number of Privilege Escalation and Impact techniques are used. In contrast, if a large number of Persistence

| Sorebrect | WannaCry | Magniber | Emotet | Gandcrab |
|---|---|---|---|---|
| Spearphishing attachment, Link, Via service | - | Drive-by compromise | Spearphishing attachment | Drive-by compromise |
| Service execution | Windows management instrumentation | Scripting, Exploitation for client execution, Scheduled task, Windows management instrumentation | User execution, Command line interface, Scripting, PowerShell | Scripting, PowerShell, Windows management instrumentation |
| - | New service, Registry run keys | - | New service | - |
| Process injection | New service | Process injection | - | Process injection |
| Process injection, Indicator removal on host | Hidden files and directories, File permission modification | Obfuscated files or information, Process injection | Obfuscated files or information, Masquerading | Obfuscated files or information, Deobfuscate/ Decoded files or information, Process injection, Disabling security tools |
| Brute force | - | - | Credential dumping, Credential in files, Brute force | - |
| Network share discovery | Network share discovery, File and directory discovery | File and directory discovery | Process discovery | Process discovery |
| - | Exploitation of remote services, Remote file copy | - | Exploitation of remote services | - |
| - | - | - | Data from local system, Email collection | - |
| Multi-hop proxy, Multilayer encryption | Multi-hop proxy, Multilayer encryption | Remote file copy | Remote file copy, Commonly used port, Uncommonly used port, Data encoding, Data obfuscation | Remote file copy |
| - | - | - | Exfiltration over command and control channel | Automated exfiltration, Exfiltration over alternative protocol |
| Inhibit system recovery, Data encrypted for impact | Service stop, Data encrypted for impact, Inhibit system recovery | Data encrypted for impact, Inhibit system recovery | - | Data encrypted for impact, Inhibit system recovery |

and Defense Evasion techniques are used to prevent malware from being detected or for extending its life, for example, we classify the attacks as "evasive fileless" cyberattacks.

Finally, Credential Access, Discover, Collection, and Exfiltration stages form a collection of techniques that malware uses to obtain accounts or certain information from a user's PC and leak the information. Therefore, when Credential Access, Discover, Collection, and Exfiltration are discovered, we classify these attacks as "collective fileless cyberattacks." The techniques of each stage identified using the MITRE ATT&CK TTP information provided by Cuckoo Sandbox are presented in Table 5. This table also shows that the number of techniques used by each attack category for each file is less than that of the cyberattacks. Based on these results, we can confirm that the techniques used by malicious code such as Poweliks, Kovter, and WannaCry can be described as evasive fileless cyberattacks.

Furthermore, Kovter is a similarly constructed malicious code based on Poweliks, and it can be observed that such categorization can be helpful for classification based on the characteristics of malicious code. In addition, information obtained by cyberattacks such as Sorebrect has indicated that approximately 33% of attacks for all six techniques are collection-based cyberattacks. In the case of WannaCry, two Collection techniques are used in the same way as Sorebrect, although this tendency is not significant because this category
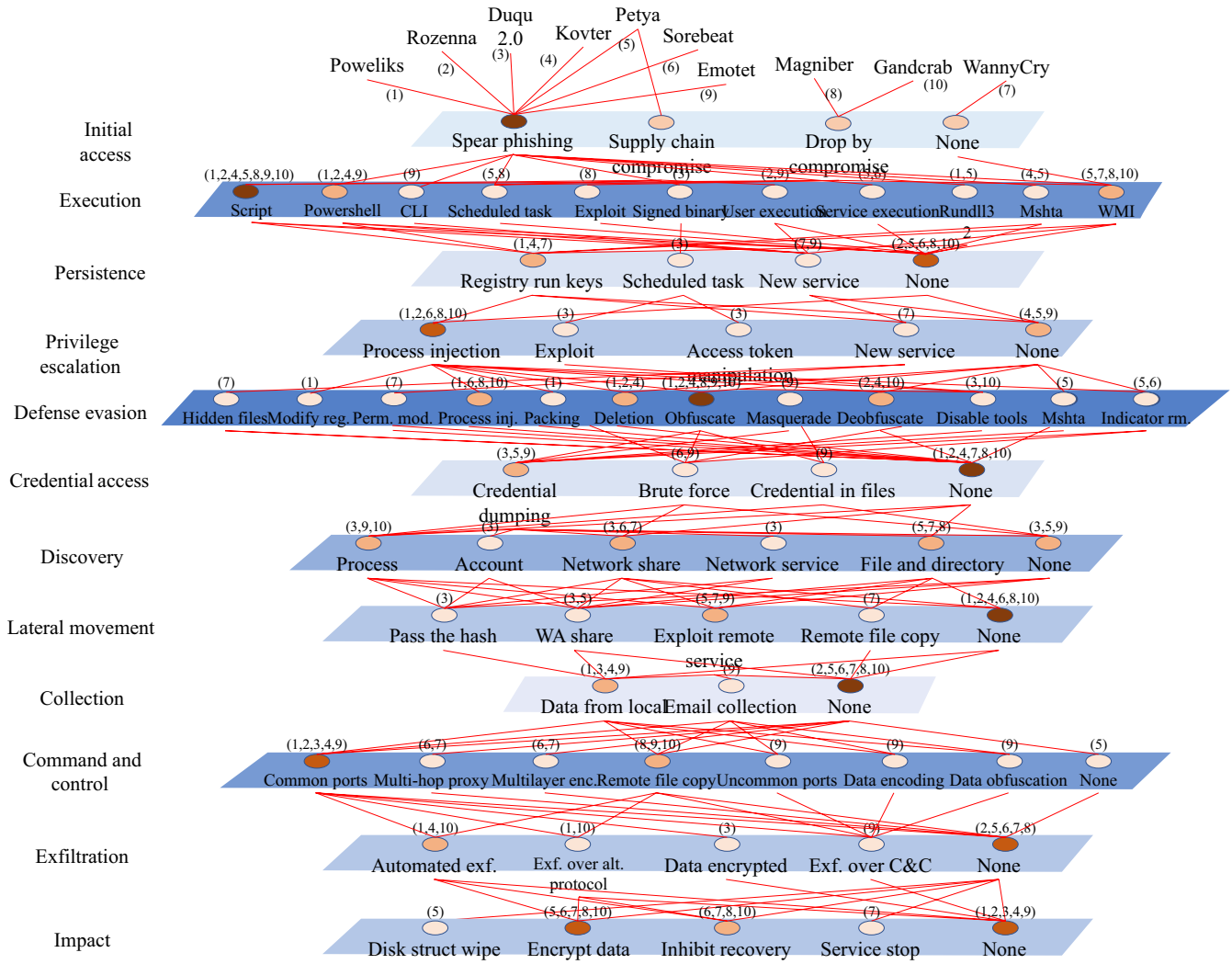
**FIGURE 2** Map of the links between fileless cyberattack stages and techniques

is low for the full technology. However, because malware usually has more than one purpose, the malware has not been classified as a certain type. Instead, using numeric values, we strive to identify how close the malware is to each classification.

To achieve this, Algorithm 1 divides the number of techniques used in each type by the total number of techniques, thereby confirming the ratio of techniques in the three categories of fileless malware. Algorithm 1 generates the graphs that classify malware into different types using the previously generated MITRE ATT&CK stage information for each malware. In addition, it reveals the characteristics of each attack category (Evasion, Attack, and Collection) of cyberattacks using the information in Table 5. It also uses the signature information generated from the Cuckoo Sandbox result and measures the number of techniques applied in each category for each malicious code. Subsequently, the ratio is calculated by dividing the number of used methods for each category by the total used techniques of each malicious code type. Based on the calculated ratio, Algorithm 1 generates a simple 3D-based graph (using Table 5), as shown in Figure 3.

| Algorithm 1 **Fileless cyberattack classification and visualization** |
|---|
| 1:     **procedure** MAKE_GRAPH(*input_file, output_file*) |
| 2:         data = |
| 3:         **for** *line* in *input_file* **do** |
| 4:             **for** *i* in range(0, 12) |
| 5:                 add line[*i* + 1] in data |
| 6:             **end for** |
| 7:         **end for** |
| 8:         *class_data* = |
| 9:         **for** *mal_info* in *data* **do** |
| 10:            *class_data*[*Evasion*] =number of Evasion type signatures in *mal_info* |
| 11:            *class_data*[*Attack*] =number of Attack type signatures in *mal_info* |
| 12:            *class_data*[*Collection*] =number of Collection type signatures in *mal_info* |
| 13:            *class_data*[*Others*] =number of other signatures in *mal_info* |

**TABLE 5** Cuckoo sandbox TTP matching information

| Cyber attack | EVASION | | ATTACK | | COLLECTION | | | | Total |
| | Persistence | Defense Evasion | Privilege Escalation | Impact | Credential Access | Discovery | Collection | Exfiltration | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Poweliks | 4 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 11 |
| Rozena | 3 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 10 |
| Duqu 2.0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| Kovter | 4 | 5 | 2 | 0 | 1 | 4 | 0 | 0 | 16 |
| Petya | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| Sorebrect | 1 | 2 | 1 | 0 | 0 | 2 | 0 | 0 | 6 |
| WannaCry | 6 | 4 | 2 | 0 | 0 | 2 | 0 | 0 | 14 |
| Magniber | 2 | 2 | 2 | 0 | 0 | 1 | 0 | 0 | 7 |
| Emotet | 3 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 9 |
| GandCrab | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 5 |

| Algorithm 1 **Fileless cyberattack classification and visualization** |
| --- |
| 14:     **end for** |
| 15:     $x\_coords = class\_data[\text{Evasion}]/(class\_data[Evasion]$ $+ class\_data[Attack] + class\_data[Collection])$ |
| 16:     $y\_coords = class\_data[\text{Attack}]/(class\_data[Evasion]$ $+ class\_data[Attack] + class\_data[Collection])$ |
| 17:     $z\_coords = class\_data[\text{Collection}]/(class\_data[Evasion]$ $+ class\_data[Attack] + class\_data[Collection])$ |
| 18:     $make\_graph(x\_coords, y\_coords, z\_coords)$ ▷Make 3D graph using $x$, $y$, $z$ coords as axes |
| 19:     **end procedure** |

From Figure 3, the types of characteristics of and the techniques mainly included in each type of malware can be extrapolated.

# 4 | DISCUSSION

In the case of a fileless cyberattack, the code for conducting malicious actions is the same as that of traditional malicious code. However, there is one major difference with fileless malicious code in that, unlike traditional malware, fileless cyberattacks do not attempt to leave files on the victim's PC. This means that even if the victim is attacked by malicious code, the attack cannot be identified by a file stored on the PC. To this end, fileless cyberattacks are executed in the form of scripts through programs such as PowerShell and WMI, and the scripts are embedded in the memory and registry information. In addition, the code execution is registered in the same place as the task scheduler, and some fileless cyberattacks also remove all files they used to further conceal the evidence of the attack. Consequently, victims and analysts have difficulty obtaining information about the malicious

code from the PC and obtaining samples for fileless malicious code analysis.

During the experiment, the first stage was the use of Cuckoo Sandbox, which was applied for convenience. To evaluate the malicious code analysis results generated by Cuckoo Sandbox, the important criterion was whether the information about the signature was sufficient to be used in the actual analysis. The results of our analysis differ from those of a published report. Table 4 summarizes the techniques used, and Table 5 summarizes the techniques considering the results of the Cuckoo Sandbox analysis. However, we determined that this difference was not likely to affect the use of the Cuckoo Sandbox results. Because there are some analyses that are not included in the TTP of Cuckoo Sandbox, additional research is required to consider this issue further.

We identified the attack techniques implemented by the fileless cyberattack using Cuckoo Sandbox and classified these techniques based on the MITRE ATT&CK framework. Subsequently, the types of fileless cyberattacks identified were divided into three categories (Evasion, Attack, and Collection), and the number of attack techniques used by each fileless cyberattack was investigated. In the case of Poweliks, approximately 20%, 70%, and 10% of all attack techniques are related to the Attack, Evasion, and Collection categories, respectively. Based on this categorization, it can be observed that a Poweliks fileless cyberattack uses malicious code based on an evasion, and a method for collecting user information is not applied. This analysis confirms the main purpose of the considered fileless cyberattacks.

Some studies on existing malicious code classification methods have not considered fileless cyberattacks. In contrast, this paper discusses how to classify malware by focusing on such attacks, which is a relatively recent issue. In addition, instead of using the results of Cuckoo Sandbox, we analyzed the results more closely and prepared materials that
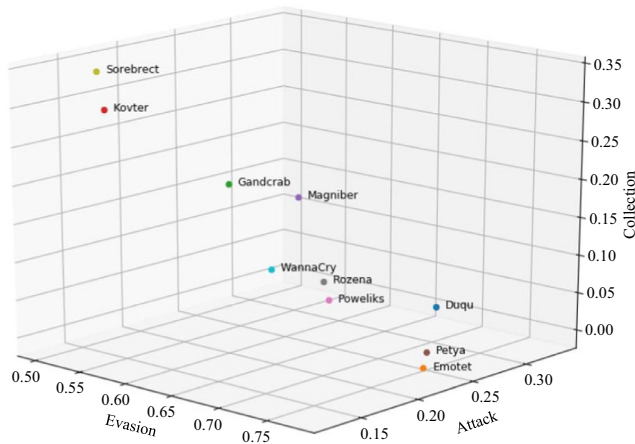
**FIGURE 3** Results of fileless malware classification three categories

can be viewed as the basis for classification. In addition, if the method for classifying existing fileless cyberattacks is a part of the attack technique used by malicious code, such as the use of memory and a script, classification was conducted in this study over a wide range based on a large number of attack techniques. Because the attack techniques used for classification described in this paper are largely categorized into eight stages, it is possible to classify such attacks more closely than with existing methods that focus on only one stage. In addition, the characteristics of a file with less malicious code were quantified to classify the extent of the unique characteristics of the code.

It was also confirmed that all 10 analyzed fileless malicious code types used more than half of the evasion techniques. As such, it was confirmed that fileless cyberattacks use multiple attack techniques to avoid detection. By checking the ratio of the three categories for each malicious code, it appears to be straightforward to immediately determine the techniques on which the malicious code focuses.

# 5 | CONCLUSION

As cyberattacks continue to advance and become more complex, the techniques used to detect and prevent such attacks are also steadily developing. Furthermore, fileless cyberattacks continue to bypass malware detection techniques.

This study analyzed 10 fileless cyberattacks that have recently emerged. The analysis of these cyberattacks revealed the characteristics and specific techniques used. In addition to an analysis of published reports on fileless cyberattacks, actual samples were obtained and analyzed using Cuckoo Sandbox.

By dividing the number of each type of technique used in a specific fileless cyberattack by the total number of techniques

available, each ratio was identified and analyzed across three dimensions. Through this process, fileless cyberattacks were classified into the following categories: Evasion, Attack, or Collection. Through this research, we expect to provide a foundational framework for identifying and classifying the characteristics of fileless cyberattacks that are likely to emerge in the future.

## ORCID

*Kyounggon Kim* https://orcid.org/0000-0002-5675-4253

## REFERENCES

1. S. Herzog, *Ten years after the Estonian cyberattacks: Defense and adaptation in the age of digital insecurity*, Georgetown J. Int. Affairs, **18** (2017), 67-78.
2. J.-Y. Kong, J. I. Lim, and K. G. Kim. *The all-purpose sword: North korea's cyber operations and strategies*, in Proc. Int. Conf. Cyber Conflict (Tallinn, Estonia), May 2019, pp. 1–20.
3. K.-G. Kim, *State-sponsored hacker and changes in hacking techniques*, 2017.
4. F. Dang et al., *Understanding fileless attack on linux-based IoT devices with HoneyCloud*, in Proc. Annu. Int. Conf. Mobile Syst., Applicat., Services (Seoul, Rep. of Korea), June 2019, pp. 482–493.
5. B. N. Sanjay et al., *An approach to detect fileless malware and defend its evasive mechanisms*, in Proc. IEEE Int. Conf. Computational Syst. Inf. Technol. Sustainable Solutions (Bengaluru, India), 2018, pp. 234–239.
6. B. S. Rivera and R. U. Inocencio, *Doing more with less: A study of file less infection attacks*, Virusbulletin, (2015).
7. Sudhakar and S. Kumar, *An emerging threat fileless malware: A survey and research challenges*, Cybersecurity. **3** (2020), 1–12.
8. The evolution of the fileless click-fraud malware poweliks, https://www.symantec.com/content/dam/symantec/docs/securitycenter/white-papers/evolution-of-fileless-click-fraud-15-en.pdf, Accessed: 06.09.2015
9. G. Lee, K. Kim, and S. Lee, *Analysis and detection methods for the fileless in-memory malwares*, 2017 Conference on Information Security and Cryptography-Summer, 2017.
10. B. Mo et al., *The classification model of fileless cyber attacks*, J. KIISE **47** (2020), 454–465.
11. Paul Rascagnères, *Poweliks: The persistent malware without a file*, 2016.
12. GData, *Where we go, we don't need files: Analysis of fileless malware "rozena"*, https://www.gdatasoftware.com/blog/2018/06/30862-filelessmalware-rozena, Accessed: 08.03.2020
13. Z. Kim, *Attackers stole certificate from foxconn to hack kaspersky with Duqu 2.0*, Wired, June 2015.
14. Check Point, *Kovter ransomware – the evolution: From police scareware to click frauds and then to ransomware*, https://blog.checkpoint.com/2016/04/15/kovter-ransomware-theevolution-from-police-scareware-to-click-frauds-and-then-toransomware/, Accessed: 08.03.2020
15. CISA, *Petya ransomware*, https://www.uscert.gov/ncas/alerts/TA17-181A, Accessed: 08.03.2020
16. A. Berry, J. Homan, and R. Eitzman, Wannacry malware profile, Hentet fra, https://www.fireeye.com/blog/threatresearch/2017/05/wannacry-malware-profile.html, 2017.

17. MalwarebytesLab, Magniber ransomware: Exclusively for south koreans, https://blog.malwarebytes.com/threatanalysis/2017/10/magniber-ransomware-exclusively-for-southkoreans/, Accessed: 08.03.2020

18. AhnLab, *Asec report vol.88 q3 2017*, https://global.ahnlab.com/global/upload/download/asecreport/ASECREPORT_vol.88_ENG.pdf, Accessed: 08.03.2020

19. AhnLab, *Asec report vol 91 q2 2018*, https://global.ahnlab.com/global/upload/download/asecreport/ASECREPORT_vol.91_ENG.pdf, Accessed: 08.03.2020

## AUTHOR BIOGRAPHIES
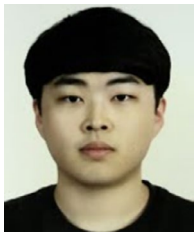
**GyungMin Lee** received his BS degree in computer science from Korea University, Rep. of Korea, in 2018. He also received his MS degree at the Graduate School of Information Security, Korea University, Rep. of Korea, in 2020. His research interests include information security, system security and online game security.

**ShinWoo Shim** received his BS degree from Pohang University of Science and Technology, Rep. of Korea, in 2007. He also received his MS degree from Korea University, Rep. of Korea, in 2019. His research interests include cyber command and control, mission impact assessment and cyber threat response.

**ByoungMo Cho** received his BS degree from Inha University, Rep. of Korea, in 2001. He also received his MS degree from Inha University, Rep. of Korea, in 2003. His research interests include cyber security, modeling and simulation.

**TaeKyu Kim** received his BS degree from Chung-Ang University, Rep. of Korea, in 2000. He also received his MS degree from the University of Arizona, USA, in 2006 and received his PhD degree from the University of Arizona, USA, in 2008. Currently, he is the leader of the cyber-warfare team at the LIG Nex1 Intelligent SW Research Center. His research interests include cyber command and control, mission impact assessment and cyber threat response.

**Kyounggon Kim** received his BS degree in computer science from Soongsil University in 2008, and his MS and PhD degrees in School of Cybersecurity from Korea University, Rep. of Korea, in 2015 and 2020, respectively. He is an assistant professor at Naif Arab University for Security Sciences. He has performed penetration testing for over 130 clients in various industries when he worked for Deloitte, PwC, and boutique consulting firms for over 15 years. He was awarded the 6th place at DefCon CTF in 2007 and a first prize at the First Hacking Defense Contest hosted by the Korea Information Security Agency. He has authored a book on internet hacking and security and has translated numerous security books. His research interests include vulnerability analysis, smart city security, and cyberphysical systems and Internet-of-Things security.