IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2020, accepted December 5, 2020, date of publication December 10, 2020,
date of current version December 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043939

A Study on the Digital Forensic Investigation Method of Clever Malware in IoT Devices

DOHYUN KIM[1], YI PAN[2], (Senior Member, IEEE), AND JONG HYUK PARK[3], (Member, IEEE)

[1]Department of Computer Engineering, Catholic University of Pusan, Busan 46252, Republic of Korea
[2]Department of Computer Science, Georgia State University, Atlanta, GA 30302-5060 USA
[3]Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul 01811, Republic of Korea

Corresponding author: Jong Hyuk Park (jhpark1@seoultech.ac.kr)

This work was supported by the Advanced Research Project funded by the Seoul National University of Science and Technology (SeoulTech).

ABSTRACT As IoT devices are always connected to mobile devices or other computing devices via the Internet, clever malwares targeting IoT devices or other computing devices connected to IoT devices are emerging. Therefore, effective IoT security research is needed to respond to hacking attacks by these kinds of malware. This paper studied the method of identifying and analyzing malware combined with social engineering from the perspective of digital forensics. The paper classified and analyzed intelligent malware characteristics and proposed a method of quickly identifying and analyzing the malware that secretly intruded into the devices installed with Android, Linux OS, using digital forensics techniques. Moreover, this paper proved its effectiveness by applying this investigation method to two actual malware cases. The research outcomes will be useful in responding to increasingly clever malware attacking IoT devices.

INDEX TERMS IoT security, IoT device forensics, IoT malware, malware investigation, social engineering malware.

I. INTRODUCTION

Many people today use computers and mobile devices such as smartphones, tablet PCs, smartwatches, smart cameras, navigation systems, and IoT devices such as smart TVs, AI speakers, robot vacuum cleaners, and various other home networking devices in their daily lives. Even electric cars like Tesla, which can be considered an IoT device, have recently been connected to the network. The number of IoT devices owned by individuals continues to increase; in fact, 13.6 IoT devices are expected to be owned per US citizen by 2022 [1].

As these various IoT devices are closely used in everyday life, various kinds of information are stored. In general, private information such as call history, messages, photos, and videos is stored in these embedded devices. Moreover, with the recent release of various health services and apps available on wearable devices, vital personal biometric information can also be stored.

In addition to this, the recent increase in telecommuting due to COVID-19 and the trend of Bring Your Own Device (BYOD) have led to a lot of work-related information being stored inside these devices. Therefore, there is a steady increase in malware to attack IoT devices, smartphones, and wearable devices [2], [3]. Since they are becoming more intelligent combined with social engineering techniques, research on the prevention of malware and incident response in the IoT environment is needed [4].

We studied the incident response from a digital forensic perspective to detect and analyze effectively malware that use various social engineering techniques; the target of attacks was Android OS, which has the highest share of the global mobile operating system market (72.26%) [5] between 2019 and 2020. Intelligent malware, which uses social engineering techniques, generally has three characteristics that make incident response difficult:

First, malware combine various social engineering techniques, such as phishing and smishing, to break into the device through various methods. Even after a successful intrusion, they continue to attack, such as phishing and vishing, to steal the personal and financial information of the device users. Therefore, it is difficult for investigators to analyze when and by what route this malware broke into the device. Second, malware also disguises itself as benign or downloads and installs additional malware in the process of breaking into the device and attacking it. Moreover, this lure users to delete the antivirus app, making it difficult for

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Lopez.

investigators to analyze which of the apps installed on the device are malware or benign. And third, Malware mainly operates in the background and leaks users' private and financial information to a command and control server (C&C server) created by hackers to gather information from victims. Therefore, it is difficult to trace back a real hacker.

Due to the characteristics above, the malware also leaves traces of file systems, system logs, and app logs when downloaded, installed, and operated on devices. Therefore, considering the characteristics of IoT malware combined with social engineering techniques and the method of investigating only existing app installation files, malware can be detected very effectively from a digital forensic perspective.

To this end, we analyzed the actual cases of accidents caused by malware that use social engineering techniques and leak various kinds of personal and financial information. As a result, we studied the investigation model for effective incident response from a digital forensic perspective. Our studies and contributions are as follows:

1) We propose the types of artifacts necessary for the investigation, such as file systems, system logs, app logs, etc. where traces of malware inevitably remain, including how to analyze them. As such, we contribute to suggesting new types of files needed to investigate malware that has not been studied before and to conduct efficient research.
2) We study how to analyze when and how malware penetrated devices by reversing the characteristics of social engineering techniques that can be used by malware. Because malware uses apps such as SMS, messenger, and web browser to penetrate devices, we contribute to finding out much information that is hard to know by analyzing the behavior of the malware's installation files.
3) We study how to analyze the traces and timing of the installation of malware after it infiltrates the device. Even if the malware was downloaded to the device, it must be installed to perform malicious acts, making users use a phishing unknowingly. We can find out when and how users were tricked by malware into allowing it to be installed through our research.
4) We study the factors and methods of analysis that must be investigated to analyze the malware's malicious behavior. The malware that first penetrated the device often gets installed simply as a dropper and takes steps for the actual performance of malicious behavior. We propose a way to analyze these behaviors effectively.
5) We suggest how malware analyzes information from C&C server that leak information from stolen users and collect the information needed to trace back the hackers or groups of hackers that created the malware. The results can contribute to analyzing the actual location of hackers or groups of hackers, when malware was created, and so on.

The rest of this paper is organized as follows: Chapter 2 introduces the background and related research on IoT malware analysis; Chapter 3 proposes a useful investigation model for malware forensics investigation on IoT devices; Chapter 4 analyzes cases wherein accidents occurred due to malware combined with social engineering techniques from a digital forensic perspective; Chapter 5 is a discussion of our research; finally, Chapter 6 presents the conclusion and future work.

II. LITERATURE REVIEW

A. CHRACTERISTICS OF IoT MALWARE

Intelligent malware, which threatens IoT devices' security, began to emerge as an issue around 2013, and most of them targeted smartphones with Android and iOS software. These are mainly combined with social engineering techniques such as Phishing, Smishing, and Vishing in order to engage in malicious behavior that intrudes on the devices, obtain administrator privileges, or leak user information to hackers [6].

1) Phishing attack: This attack encourages the victim to access the hacker-modified web page directly or automatically without any doubt and then enter and submit his/her personal and financial information or download and install malware on his/her IoT devices through a drive-by download attack. Hackers typically use phishing techniques on malware that are downloaded and installed to trick the user into thinking that the app is benign. The malware acts as a dropper and additionally downloads and installs another malware or impersonates a financial app, requiring victims to enter their personal or financial information.
2) Smishing attack: This is an abbreviation for SMS phishing, one of the phishing attacks that use services that allow hackers to send messages such as text messages as well as on messengers and social media apps to induce victims to access malicious web pages. The hacker sends to the victim a message that is a social issue or a message that the victim might be interested in and links to malicious web pages to induce the victim to click it. If the victim accesses a malicious web page through a link, the hacker steals the victim's information in the same way as the conventional phishing technique or allows the malware to be downloaded to the victim's device automatically.
3) Vishing attack: This is an abbreviation for voice phishing, one of the phishing attacks wherein hackers call victims directly and impersonate investigative agencies or financial institutions to obtain personal information and financial information.
4) APT attack: This is an attack that targets IoT devices of specific targets, not an unspecified number but by properly mixing Phishing, Smishing, and Vishing techniques. To target the final target, it sometimes targets other IoT devices around the victim first and attacks the target IoT devices based on this. This attack can take a

long time, but it is compelling because it attacks very carefully.

B. STUDIES ON MOBILE MALWARE ANALYSIS AND DETECTION: STATE-OF-THE-ART AND TRENDS

Since most IoT devices can use apps such as web browsers and messengers, the same methods of malware used in smartphones can be applied to various IoT devices; hence the need for the analysis of the existing mobile malware.

Several studies have been conducted to analyze and classify various phishing techniques that can occur on mobile devices [7]–[10], including those carried out to analyze and classify attacks on social engineering techniques [11]. Several studies have been conducted to cope with the growing damage caused by the prevalence of smishing malware. Some of them used the Naive Bayesian classifier to detect suspected smishing [12], [13], which analyzes the characteristics of smishing characters and detects smishing characters using rule-based methods [14]. In addition, research was conducted to detect malware by analyzing the network traffic of apps [15] and combining permission information and API call [16].

Studies have been conducted to detect malware dynamically by applying taint analysis to Android malware [17]–[19]. Likewise, studies have been conducted to detect malware by analyzing the behavior of the app through data flow analysis [20]–[23]. There was also a study that comparatively analyzed the research results of the existing data flow analysis [24].

In order to analyze malware more effectively, PACE, an integrated solution that provides machine learning-based Android malware detection technology using REST API, web interface, and ADB interface, has been proposed [25]. Moreover, to overcome the shortcomings of malware detection in Android emulators, studies were conducted on dynamic analysis using machine learning for malware detection on real devices [26], [27].

There have been studies that proposed a method of detecting malware by applying data mining techniques to a signature-based, motion-based detection [28]. There was also a malware detection study based on permission usage analysis by mining information on the permissions of Android apps [29]. In this study, Significant Permission Identification (SigPID) was developed. SigPID detects malware by mining information on permissions; as a result of the experiments, it effectively detects new malware. Moreover, there was a study that developed an engine called DroidDetector to improve the detection function of malware through the effective extraction of the characteristics of malware by combining the features of static analysis and dynamic analysis of Android apps with deep learning technology [30]. There was also a study proposing a mobile forensic platform that detects and analyzes Android malware called ToR-SIM Platform [31].

C. DIGITAL FORENSIC INVESTIGATION FOR INCIDENT RESPONSE

There was a study that classified and analyzed representative cyber-attacks that occurred in industrial control systems and presented an incident response model in this environment [32]. Another study – in order to investigate cyber-attacks – proposed a digital forensics framework by applying detailed instructional steps in the data inspection and analysis stages and demonstrated this through a real-world example of D4I [33]. Moreover, studies have been conducted to detect threats automatically and respond effectively to cyber-attacks on the Cloud storage system [34]. The study confirmed that its results could be used effectively in Amazon Web Services (AWS) and Google Cloud Platform (GCP). Moreover, studies have suggested methods for detecting Ransomware attacks and investigating digital forensics [35].

Research on malware in IoT devices from a digital forensic perspective has not been conducted much, however. As IoT devices are widely used in everyday life, research is needed to effectively investigate the malware that threatens them.

III. PROPOSED METHODOLOGY

In this chapter, we propose an investigation model, as shown in Figure 1, for an effective incident response that can detect, analyze, and track intelligent IoT malware combined with social engineering techniques from the perspective of a digital forensic investigation. Generally, people install and use numerous apps on IoT devices. In particular, on average, people in the United States, South Korea, and Japan install more than 100 apps and use 30 to 40 of them a month [36]. Therefore, it takes too much time to analyze all apps' behavior on IoT devices to detect malware.

For efficient incident response, we first analyze file system information, system log, and specific app log that inevitably leaves traces of IoT malware and investigate whether malware is installed or not. We then analyze how the malware broke into IoT devices as well as its malicious behavior including information on malware. The proposed investigation model consists of five phases (PHASE 1, 2, 3-1, 3-2, 4) and a total of seven phases in detail, and the contents for each step are as follows:

1) PHASE 1. Preprocessing: This step involves selecting and extracting the files to analyze essential information for investigation. These files, in detail, are equivalent to Table 1.

- File containing the file system information of all files in the /sdcard area from the flash memory of IoT device: This file contains the file system metadata information (filename, size, generation time, modification time, etc.) of all files in the /sdcard area. Malware, which penetrates the device with social engineering techniques such as phishing and smishing, inevitably has to download its installation files to the /sdcard area. Therefore, this file is required to analyze the presence of app installation files in the /sdcard area. For Android,

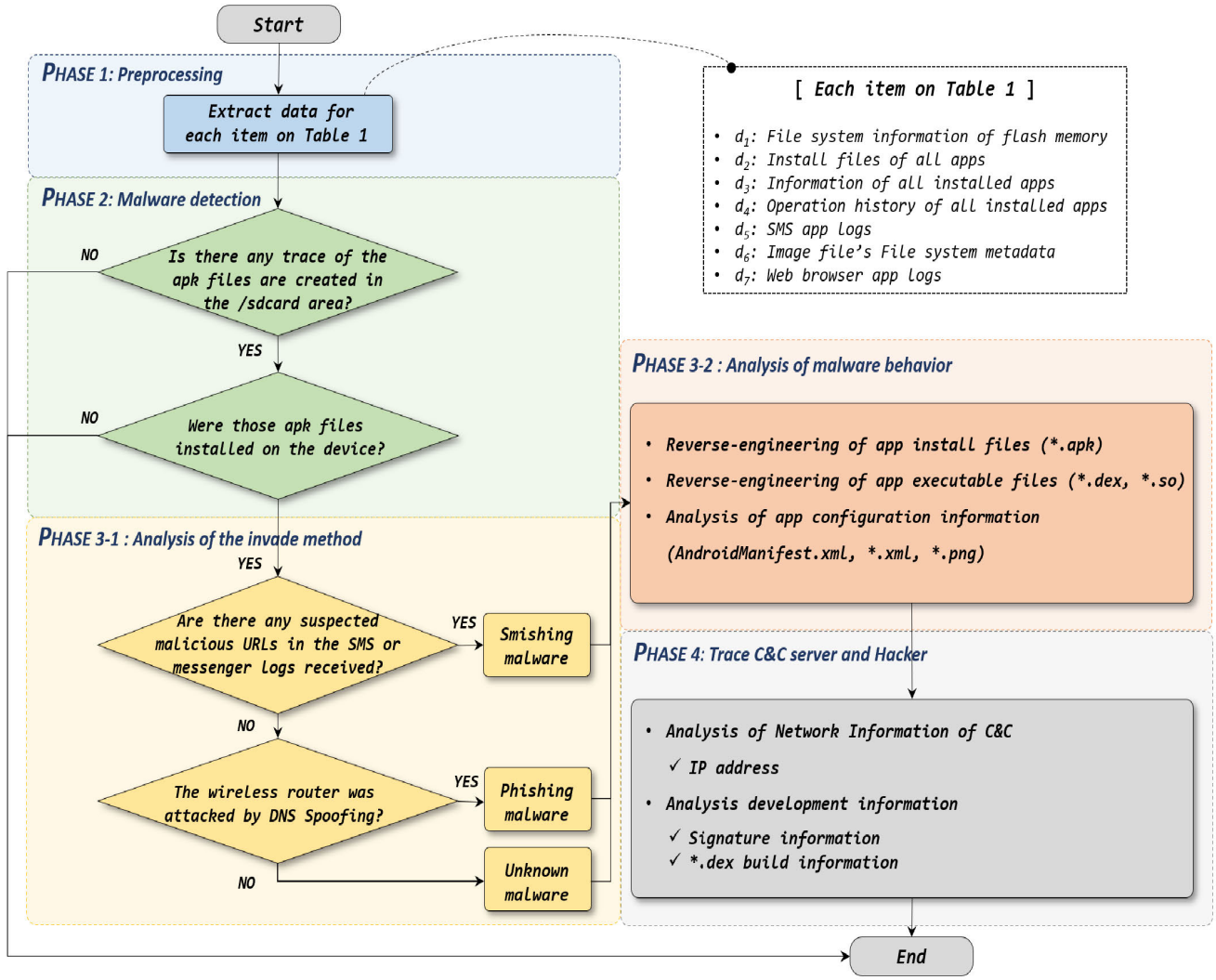


FIGURE 1. Forensic investigation model for malware of IoT device.

TABLE 1. Analysis target data.

No	File Path	Description
d_1	/data/data/com.android.providers.media/databases/external.db	File system information of flash memory (e.g., file system metadata such as file path, name, size, creation, and modification timestamp, etc.)
d_2	/data/app/*.apk, /data/media/*/*.apk	Install files (apk) of all installed apps
d_3	/data/data/com.android.vending/databases/localappstate.db	Information of all installed apps (e.g., package name, app name, download and update timestamp, Google Play account, etc.)
d_4	/data/system/dmappmgr.db	Operation history of all installed apps (e.g., last launch, pause, service start, service stop timestamp, total usage time, launch count)
d_5	/data/data/com.android.provider.telephony/databases/mmssms.db	SMS app's usage history (e.g., sent & received timestamp, message, phone number, etc.)
d_6	-	EXT4 File system metadata (e.g., inode table, directory entry, etc.)
d_7	/data/data/com.nhn.android.search/databases/search.db	Web browser app's usage history (e.g., Naver browser)

this is the file corresponding to d_1 in Table 1. If these files do not exist, the investigator can analyze the metadata of the file system instead, such as inode table, and directory entry of the EXT file system.

- Installation files of all apps inside the IoT device: These are the files corresponding to d_2 of Table 1, which the investigator selects and analyzes in detail among files suspected to be malicious.

- File containing the metadata of apps installed on IoT devices: This file corresponds to d_3 in Table 1; for Android, it contains information such as the app's package name, app name, download time, installation time, last update time, and account used for download on Google Play.
- File containing the execution history of the components of the apps installed on the IoT device: This file is equivalent to Table 1's d_4, containing information such as the last launch time, end time, background operation time, total launch count, and total usage time.
- Logs of all apps installed on IoT devices that can send and receive messages: These files correspond to d_5 in Table 1. They include details of users' history of SMS apps, social media apps such as Facebook or Instagram, and messenger apps such as Telegram, Naver Line, or Kakao Talk.
- Metadata area of the file system on IoT device: This data corresponds to d_6 in Table 1, such as inode table, directory entry, e.g., the EXT file system.
- Log of all apps installed on IoT devices for web browsing: These files correspond to d_7 in Table 1, storing the history of apps that offer web browsing features such as Chrome, Samsung browser, and Naver browser.

2) PHASE 2. Malware detection: This step involves checking for the presence of malware inside IoT devices. Apps downloaded to IoT devices through smishing and APT attacks without going through official app stores such as Google Play will generate installation files of the app in the /sdcard area. Therefore, the investigator can quickly detect malware by checking for the existence of the app installation file in the /sdcard area and analyze its malicious presence as applicable.
   In this process, the following additional files can be analyzed for malware detection analysis. For detailed file system analysis, the metadata (d_6: Inode, Directory Entry, etc.) of the EXT4 file system and the Android system log (d_1: external.db) can be analyzed. Likewise, for the detailed analysis of apps installed on IoT devices, a file containing installation and update time (d_3: localappstate.db) as well as files where the actions of the apps are stored (d_4: dmapmgr.db) can also be analyzed.
   By analyzing these files, investigators can analyze when each app was installed and updated, and the accounts of Google Play that users used to install each app.
3) PHASE 3-1. Analysis of the invade method: This step is to analyze how the malware identified in PHASE 2. has invaded the smartphone.
   If a text message containing the URL was received and then accessed the URL's web page via a web browser or web view, and the app installation file, an apk file, was created in the /sdcard area, this can be analyzed as smishing malware. And if the DNS server on the wireless router is tampered with and after accessing a specific web page, the app installation file is created in the /sdcard area; this is the phishing malware. This is discussed in 4.2 in more detail in a practical case.
   As a result of this analysis, the investigator can find out how malware are used to intrude on IoT devices (e.g., click malicious links included in SMS messages or automatically access specific web pages due to phishing) as well as their download methods (e.g., drive-by download).
4) PHASE 3-2. Analysis of malware behavior: This step involves analyzing the behavior of the malware found in PHASE 2. For Android, tools such as JEB are used to analyze executable files with built-in JAVA language [37], or tools such as IDA PRO are used to analyze library files in the ELF file format 38]. Moreover, the AndroidManifest.xml file can be used effectively to analyze the app's permission and settings information.
5) PHASE 4. Trace C&C server and Hacker: This step is to track where information was leaked due to malware, and the analysis results of PHASE 3-2 can be used. This analysis step allows investigators to find the information needed to track hackers or groups of hackers who have created and used malware.
   The investigator can get information from the C&C server by analyzing the behavior of malware sending out information, such as users' personal and financial information. In addition, analyzing the signature information of the app and the build information of the executable $*$.dex enables finding out when the corresponding malware was created. If a large quantity of malware is collected, it can also be used to group the malware.

IV. EXPERIMENTAL RESULTS

In this chapter, we make an incident response from a digital forensic perspective in two cases wherein an accident occurred due to malware combined with social engineering techniques. These cases are actually cases wherein we have been asked to investigate the forensics. There are two types of cases: the combination of smishing and vishing and the combination of phishing and APT attack. In particular, the second case involves a hacker first attacking a wireless router in the victim's home to attack the victim's smartphone and then planting the malware into the smartphone through phishing.

A. MALWARE OF SMISHING & VISHING

In this case, hacker H steals victim A's accredited certificate and various kinds of financial information inside A's smartphone. After that, H secured additional financial information through voice phishing and borrowed money from S Bank as collateral for the money deposited in A's bank account. Furthermore, H transferred the money deposited in each account of A to his/her account. A asked us to analyze whether the malware was installed on his/her smartphone, and if so, when,

where, and how it broke into his/her smartphone and what malicious acts he/she did. Here is what A explained about the case in an interview with us:

> *A* changed his/her smartphone from Galaxy S3 to iPhone 6 on December 15, 2018. The next day, December 16, *A* installed the S-bank app version for iPhone, and the app asked *A* to enter all the numbers on the bank security card. Subsequently, on December 25, *A* received a voice phishing call from *H* impersonating an S-bank employee, and *A* was tricked into giving *H* personal information, including his/her date of birth. The next day, December 26, *H* took out a KRW 6,000,000 (about USD 4,922) loan using *A*'s name. *H* also transferred a total of KRW 9,000,000 (about USD 7,383) from *A*'s bank account to *H*'s account four times.
>
> In an interview with us, *A* said that the iPhone 6 purchased on December 15 was hacked, and that the hacking incident may have occurred through the financial information that leaked from the phone and the personal information leaked through voice phishing.

The evidence and request for analysis submitted by victim *A* are as follows:

- E_1: Apple iPhone 6 (iOS version 8.1.2), Analyzing whether there is a function that requires entering the entire number of the security card in the S Bank app and whether there is malware.
- E_2: Samsung Galaxy S3 (SHV-E210S, Android version: 4.1.2, Bootloader version E210SKSUKNK3), Analysis of malware existence.

1) FORENSIC INVESTIGATION FOR iPhone 6

Victim *A* thought that the malware was installed on E_1 (iPhone 6), so we first collected data to analyze E_1. It does not matter if the integrity of the iPhone 6 is compromised because evidence of this case is not submitted to court. Therefore, we used the jailbreaking technique to carry out data acquisition. We jailbroke E_1 using Taig with consent from A to collect data from E_1 [39]. After that, the installation file (ipa) of the S Bank app was decrypted using Clutch [30]. We then extracted the installation file (ipa) of the S-Bank app from E_1 using iTools [41] and reverse-engineered the installation file using IDA Pro.

As a result of analyzing E_1, the S-Bank app of E_1 was found to be a normal app with a function that requires the user to enter the full number of the bank security card. Therefore, we decided that E_1 contains no malware.

2) FORENSIC INVESTIGATION FOR GALAXY S3

We acquired data to analyze the Android smartphone E_2 for the second time. We imaged the flash memory of E_2 using Android Extractor with consent from A to acquire data from E_2. After that, we extracted the data corresponding to Table 1 and analyzed the data in the following five steps.

i) Search for recently installed apps: We analyzed the existence of the app installation file recently downloaded in the /sdcard area through d_1 and d6 to find traces of the malware installed on E2. Furthermore, we analyzed the types and information of recently installed and executed apps through d3 and d4. As a result of the analysis, SPAp.apk (13 December), gms.apk, and V3Plus.apk (14 December) app install files were found to have been downloaded to E2 and installed and executed. Moreover, we found that these apps were downloaded from outside, not from Google Play, and created and installed on /sdcard. We found these three files in E2 and selected them as potential malware.

ii) Trace how suspicious apps are downloaded and installed: At the time A asked us to investigate this incident, there were many incidents of smishing malware in our country, so we first analyzed d5 to investigate if there were any text messages containing URLs since December 13. As a result, we found on December 13 at 21:58:19 that the SMS "The prosecution: Report of suspects bit.ly/13jc0ms" was received on E2 and analyzed this because we were suspicious of the shortened URL.
We analyzed d_2, d_3, and d_4 to find out what web browser app A was using to access this URL. As a result, we found out that Naver's web browser was used as the web browser app. Therefore, we selected the log of this app as d_7. We analyzed through d7 that A accessed the "bit.ly/13jc0ms (original URL: http://ukk.zspoea.com/search.asp?id=98746)" included in the SMS on December 14 at 00:13:11. As a result, on December 14 at 00:13:59, the SPAp.apk, a malware installation file, was downloaded from the homepage to the /sdcard area of E2 through the drive-by download method.

iii) Analysis of suspicious apps (1/3): We reverse-engineered it using a JEB to analyze the installation files of suspicious apps in detail. SPAp.apk is a malware disguised as an app for searching for incidents through the prosecutor's office image files as in Figure 2(a). When the app is launched, a message as shown in Figure 2(b) informs the user that a "new version of the app has been released" and that the user "needs to update the Google Play service for this." After that, it shows the image as shown in Figure 2(c) to the user and installs gms.apk, another malware installation file existing in the "/res/raw/gms" path inside the installation file of this app.

iv) Analysis of suspicious apps (2/3): gms.apk is an installation file of an app that impersonates the Google Play services installed and executed from SPAp.apk. When this app is executed, it uses image files such as Figure 2(c) and Figure 3(a) to deceive the user. In addition, through the phrases in Figure 3(b), the user is informed that "the new version of AhnLab V3 Mobile

TABLE 2. Total timeline of smishing & vishing case.

No	Timestamp	Description
t_1	12-13 21:58:19	Receive smishing SMS. (from +82-10-4127-2022)
t_2	12-13 00:13:11	Access short URL included in smishing SMS through web browser app.
t_3	12-13 00:13:59	The SPAp install file, which is malware, is downloaded from the accessed website, and *A* installs it.
t_4	12-13 09:51:45	*A* runs the SPAp app, which installs and runs gms, the second malware disguised as a Google Play service.
t_5	12-14 23:33:59	The executed gms downloads and installs V3Plus.apk as the installation file of the 3rd malware disguised as an antivirus app. V3Plus is disguised as an S-bank app.
t_6	12-15	*A* changed smartphones from E_2 (Galaxy S3) to E_1 (iPhone 6).
t_7	12-16	*A* installs various apps, including financial app, at E_1.
t_8	12-18	*A* runs the S-bank app at E_2, which is actually malware V3Plus that compresses the accredited certificate in the /sdcard area of E_2 into the filename 352905050229800.zip and sends it to the C&C server.
t_9	12-25 19:34:00	*A* receives a voice phishing call and discloses his/her financial information (+ 82-2-1577-8000).
t_{10}	12-26 18:08:00	*H* took out a loan from S Bank using *A*'s name (KRW 6,000,000, USD 4,922).
t_{11}	12-26 18:11:00	Transfer from *A*'s K bank account to *H*'s account is approved (KRW 2,980,000, USD 2,444).
t_{12}	12-26 18:11:00	The second bank transfer from *A*'s K bank account to *H*'s account is approved (KRW 2,970,000, USD 2,436).
t_{13}	12-26 18:11:00	The third bank transfer from *A*'s D bank account to *H*'s account is approved (KRW 2,990,000, USD 2,452).
t_{14}	12-26 18:16:00	The fourth bank transfer from *A*'s K bank account to *H*'s account is approved (KRW 60,000, USD 49).
t_{15}	12-26 18:50:00	*A* report the crime to the police and banks.

(a)

(b)

```
<string name="app_name">검찰청 사건조회</string>
<string name="action_settings">검찰청 사건조회 Setting</string>
<string name="update_notify_title">Google Play 서비스</string>
<string name="update_notify_message">새로운 버전이 출시돼였습니다.새 버전을 설치하
<string name="spo_notify_title">경고</string>
<string name="spo_notify_message">점검중입니다.잠시후 재시도 해주세요</string>
<string name="confirm">확인</string>
<string name="cancel">취소</string>
<string name="app_down_msg">Google Play 서비스 업데이트중...</string>
```

(c)

FIGURE 2. FIGURE 2. Files inside SPAp.apk.

(a)

```
<string name="COMMON_PRODUCT_NAME">Google Play 서비스</string>
<string name="COMMON_COPYRIGHT">© 2010-2014 AhnLab, Inc. All rights reserved.</string>
<string name="notify_title">경고</string>
<string name="confirm">확인</string>
<string name="cancel">취소</string>
<string name="notify_message">새로운 버전이 출시돼였습니다.새 버전을 설치하시고 이용해주세요.</string>
<string name="app_down_msg">"업데이트 파일을 다운로드중 입니다.
설치가 차단된 경우에는 [설정]을 클릭하시고 [알수없는 소스]를 체크하신 후 어플을 다시 업데이트 하시기 바랍니다."</string>
```

(b)

FIGURE 3. Files inside gms.apk.

PLUS 2.0 vaccine[1] must be updated to use the Google Play service." After that, the malware downloads, installs, and runs the V3Plus.apk file from the webserver.

v) Analysis of suspicious apps (3/3): V3Plus.apk is a malware installation file that performs practical malicious actions to steal users' financial information. Since this malware contains image files to disguise them as various types of financial apps, it checks the types of financial apps installed on the user's smartphone, and then uses them as a disguise by displaying the same screen as the financial apps installed on the smartphone. Since the S-Bank app was installed in E_2, this malware disguised itself as an S-Bank app. The malware requests the user to input the official certificate login password, account number, account password, security card number, transfer password, etc. through the phrases in Figure 4(b) and Figure 4(c). In addition, the malware compresses the accredited certificate (/sdcard/NPKI/∗) existing in the /sdcard area of the smartphone into the filename of the smartphone device ID. Finally, the malware sends the financial information entered by the user and the compressed accredited certificate file to the C&C server. Below is the information on the C&C server that we analyzed.

- Domain name: 00.728371.com
- IP address: 119.241.210.180
- Location: Japan
- Registry domain ID: cheng hu
- Type of web server: apache/coyote 1.1 (JSP)
- Type of DBMS: MySQL
- OS: Windows server 2003 (language: Chinese)

Through the analysis results of E_1 and E_2, we found that A him/herself leaked personal information through voice phishing and that, in E_2, the malware was installed through a smishing attack and various kinds of financial information were leaked through this. Table 2 shows the overall attack contents in chronological order.

[1]AhnLab, a Korean antivirus company, has developed the V3 Mobile PLUS product, which occupies the largest share in Korea [42]. Most of Samsung's Android smartphones released in Korea have AhnLab's V3 Mobile PLUS vaccine installed.

FIGURE 4. Files inside V3Plus.apk.

B. MALWARE OF PHSHING & APT-ATTACKS

This is a case wherein hacker H installed malware on victim B's smartphone to steal financial and personal information such as accredited certificate, resident registration number, etc. and used this information to transact on a game item trading site in the name of B. In order to plant malware into B's smartphone, H first used APT attack techniques, such as modulating the DNS server by taking control of the wireless router installed in B's house. B asked us to analyze whether malware exists on his/her smartphone, and if so, when, where, and how it has invaded his/her smartphone and what malicious acts he/she has done. Here is what B explained about the case in an interview with us:

> B received an SMS related to i-PIN authentication on June 10, 2018 even though he did not apply for the issuance of i-PIN. A few days later, B suspected that his personal information might have been stolen and searched a website that was signed up in his/her name. As a result, B discovered that he was subscribed to a game item trading site that he did not sign up for and found out that someone had an item traded under his/her name. B said his/her smartphone seemed to have been infected with malicious code before June 10.

We first suspected a smishing accident and asked B if he had ever accessed a URL via SMS, but B said he had not. Therefore, we requested that the wireless router used by B at home be submitted for analysis to analyze accurately the path of infiltration by malware. The evidence and request for analysis submitted by B are as follows:

- E_1: Samsung Galaxy S3 (SHV-E210S, Android version 4.4.4, Bootloader version E210SKSUKNK3), Analysis of malware existence
- E_2: Wireless router: EFM ipTIME N6004 (OS: Linux, CPU: MIPS, Kernel Image name: n604m), Analysis of existence of hacking attacks

1) FORENSIC INVESTIGATION FOR GALAXY S3

We acquired data to analyze the Android smartphone E_1. We imaged the flash memory of E_1 using Android Extractor with consent from B to acquire data from E_1. After that, we extracted the data corresponding to Table 1 and analyzed the data in the following three steps.

i) Search for recently installed apps: In order to find the trace of malware installed in E_1, we analyzed the existence of the recently downloaded app installation file in the /sdcard area through d_1 and d_6. After that, the types and information of recently installed and executed apps were analyzed through d_3 and d_4. As a result, as shown in Table 3, we found that a total of 31 files suspected of malware installation files existed in the /sdcard area, and that only f9 of these files are installed in E1. Moreover, these app-installed files are changing the MD5 hash value and package name at regular intervals. We judged that this was intended to evade detection by antivirus engines that detect malware based on blacklists.

ii) Trace how suspicious apps are downloaded and installed: We analyzed the SMS and web browser usage history of E_1 to analyze the download source of each file in Table 3 and sorted them by timeline. After that, we analyzed the web page information accessed by SMS or which B received before and after the creation time of each file. Nonetheless, we could not find any traces of SMS reception or suspicious homepages containing shortened URLs used in general smishing. We could find out why the files in Table 3 were

TABLE 3. Information of install files of malware in the /sdcard area of E_1.

No	Created Timestamp	File Name	MD5 Hash Value	App Package Name
f_1	03-28 19:27:32	13722YSAY.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_2	03-28 19:28:00	38858SDGU.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_3	03-28 19:30:24	32166HBZH.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_4	03-28 19:30:47	27258KMYS.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_5	03-28 19:32:51	24742RDUQ.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_6	03-28 19:33:16	17952VSYV.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_7	03-28 19:44:48	29103HKFO.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_8	03-28 22:50:06	33957UWCM.apk	a9b8664d98f0cdc440016f594a0501f7	com.xm.status.listenner
f_9	03-29 13:13:28	28939WFFJ.apk	11b2b19f472104a79a34d726c455e6eb	com.xm.status.listenner
f_{10}	04-02 16:07:10	30005UBSY.apk	1ac727a4f76c484bcc31cbafce7e39b7	com.xmds.statuss.listenners
f_{11}	04-02 16:23:51	25965EYKK.apk	1ac727a4f76c484bcc31cbafce7e39b7	com.xmds.statuss.listenners
f_{12}	04-03 13:36:39	40040WKVS.apk	1ac727a4f76c484bcc31cbafce7e39b7	com.xmds.statuss.listenners
f_{13}	04-03 15:41:43	29103MZRM.apk	1ac727a4f76c484bcc31cbafce7e39b7	com.xmds.statuss.listenners
f_{14}	04-04 06:14:18	38377SDGQ.apk	1ac727a4f76c484bcc31cbafce7e39b7	com.xmds.statuss.listenners
f_{15}	04-06 08:32:47	16213SDIW.apk	4376441c82905b3b63f638df8d9e2460	kr.im.status.soft
f_{16}	04-07 17:26:48	35060QERM.apk	b85225ab91d6c1386834e3cd828082d9	kr.im.status.soft
f_{17}	04-08 03:23:43	26021OIWB.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{18}	04-08 03:25:34	20108MPGQ.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{19}	04-08 03:25:39	20108MPGQ-1.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{20}	04-08 15:35:44	40381VQMJ.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{21}	04-08 15:43:43	16882AEYF.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{22}	04-09 05:03:14	41031KYES.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{23}	04-09 17:12:35	13625HHFV.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{24}	04-10 05:03:56	10176MBUO.apk	27aad30e637bc3ea813a7e3164aa3258	kr.im.status.soft
f_{25}	04-19 08:09:00	35200ZSNH.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{26}	04-19 18:20:16	42225NFIY.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{27}	04-20 03:09:07	17777VSOX.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{28}	04-21 07:39:26	14601OQYY.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{29}	04-22 18:29:33	28611BWRW.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{30}	04-28 18:07:50	27914DCXA.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software
f_{31}	04-30 07:11:02	21970OMFM.apk	35597bb264ed19dbbff2e1e00636bbf9	com.imd.statuds.software

downloaded to E_1 after analyzing E_2. This is explained in detail later.

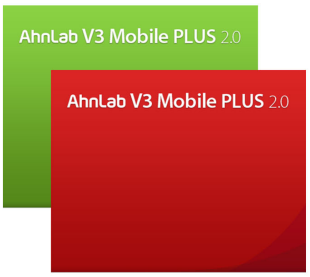
iii) Analysis of suspicious apps: We performed reverse engineering using JEB to analyze the installation files of suspicious apps in detail. As a result, it was found that all apps differ only in signature value and package name, and that the internal functions are the same apps. We analyzed in detail the f_9 of Table 3, 28939WFFJ.apk. This is malware disguised as a Google chrome web browser app, and apkprotect is applied to prevent reverse analysis; it acquires device administrator privileges during installation, hides its icon, and makes it unrecognizable by the user. When this malware is executed, it shows the image like Figure 5(a) and Figure 5(b) to the user and tricks the user as if V3 Mobile PLUS works.

After that, the malware tricks the user by disguising the malicious code as antivirus app installed on E_1 and instructing the user to delete it. The list of antivirus apps to be deleted is encoded and stored in the Config.xml file inside the app. As a result of decoding this list, it can be seen that it is a list for deleting antivirus apps from various countries as shown below.

- “id”:“103”,“value”:“com.estsoft.alyac.kt”
- “id”:“103”,“value”:“com.antivirus”
- “id”:“103”,“value”:“com.avira.android”
- “id”:“103”,“value”:“com.kisa.secheck.android”
- “id”:“103”,“value”:“com.antivirus.tablet”
- “id”:“103”,“value”:“com.inca.nprotect”
- “id”:“103”,“value”:“com.drweb”
- “id”:“103”,“value”:“com.naver.android.ncleaner”
- “id”:“103”,“value”:“com.symantec.mobilesecurity”
- “id”:“103”,“value”:“com.lookout”
- “id”:“103”,“value”:“jp.naver.lineantivirus.android”
- “id”:“103”,“value”:“com.wsandroid.suite”
- “id”:“103”,“value”:“kr.co.shiftworks.vguardweb”
- “id”:“103”,“value”:“com.TouchEn.mVaccine.web”
- “id”:“103”,“value”:“kr.co.seworks.sguard”

TABLE 4. Total timeline of phshing & apt-attack case.

No	Timestamp	Description
t_1	-	H logged into B's E_2 without authentication.
t_2	-	H modulates the DNS server address of E_2, allowing B to access his/her fake web page
t_3	03-28 19:27:32 ~ 19:44:48	B connects E_1 to Wi-Fi and accesses the Naver homepage but accesses the fake web page created by H instead. As a result, the install files of malware disguised as a Chrome web browser are automatically downloaded to E_1.
t_4	03-29 13:13:28	The install file of malware was downloaded once again to E_1 and installed in E_1.
t_5	03-29 14:01:38	The malware induces B to delete E_1's antivirus apps, creates a compressed certificate file in the /sdcard area, steals B's personal information, and sends it to the C& C server.
t_6	-	H acquires B's information from the C&C server, uses B's name to register at the game item site, and then trades the game item for others.
t_7	-	B receives the SMS related to i-PIN authentication and searches for a site registered under his/her name, suspecting that his/her information has been hacked, and finds that H has stolen his/her name and reports it to the police.



(a)

```
<resources>
    <string name="app_name">Chrome</string>
    <string name="notify">알림</string>
    <string name="notify_msg">새로운 업데이트가 있습니다. 업데이트후 이용하십시요.</string>
    <string name="confirm">확인</string>
    <string name="cancel">취소</string>
    <string name="delall">모두 삭제</string>
    <string name="security_protection">개인 정보를 보호하고 있습니다.</string>
    <string name="app1">진단명: Android-Trojan/stealer.a5a0</string>
    <string name="delall2">지금 바로 삭제 가능</string>
    <string name="app2">위치:/data/app/com.andoird.ph.network.apk</string>
    <string name="delmsg">1개의 악성코드를 발견했습니다. 악성코드를 삭제하시겠습니까?</string>
    <string name="no_sd_card">메모리 카드가 없습니다.</string>
    <string name="delete">발견한 악성코드를 모두 제거했습니다.</string>
```

(b)

FIGURE 5. Files inside V3Plus.apk.

- "id":"103","value":"com.kms.free"
- "id":"103","value":"com.estsoft.alyac"
- "id":"103","value":"com.qihoo.security"
- "id":"103","value":"com.zrgiu.antivirus"
- "id":"103","value":"com.cleanmaster.security"
- "id":"103","value":"net.btworks.phishingguard"
- "id":"103","value":"com.cleanmaster.mguard"
- "id":"103","value":"com.avast.android.mobilese"
- Etc

The malware disguised as Chrome app removes the antivirus apps installed on the smartphone; when a user accesses Naver, a fake screen is displayed to induce the user into entering personal information such as name and resident registration number. After that, the malware compresses the public certificate file (/sdcard/NPKI/∗) existing in the /sdcard area as shown in Figure 6 and sends them all to the C&C server. (IP address: 126.85.173.157, Location: Japan)

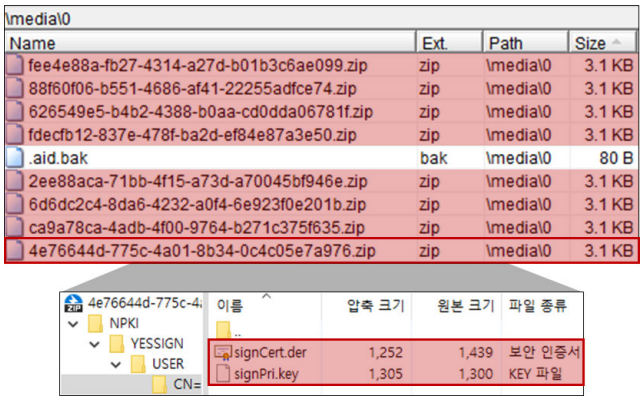


FIGURE 6. Accredited certificate files compressed by malware.

2) FORENSIC INVESTIGATION FOR EFM IPTIME N6004 (WIRELESS ROUTER)

We analyzed E_2, which is a wireless router, to analyze the inflow path of the installation files of malware that entered E_1. We imaged its flash memory through the JTAG interface. Then, using the binwalk [43], the kernel and ram disk areas were separated and analyzed from the firmware. As a result, the malware was not found in the kernel and ram disk areas, and it was found that the ID: "admin", PW: "admin", and system DNS server were manually set to the following addresses:

- Main DNS Server: 174.139.145.214 (USA)
- Sub DNS Server: 168.126.63.1

H took advantage of E_2's weak authentication security, first seized B's E_2, and tampered with the DNS server, which forced B to access the fake web page he/she wanted when using the Internet through a wireless router. Furthermore, the fake webpage forced E_1 to download malware installation files. B installed this by mistaking the malware installation file downloaded to E_1 for a normal chrome web browser, and this malware led to the removal of all antivirus apps installed on E_1 with device administrator privileges. After that, the malware stole personal information by having B enter his/her

name and resident registration number when accessing the Naver portal website, compressing the accredited certificate installed in E_1, and leaking it to the C&C server. H registered at the game item trading site using B's personal information and accredited certificate information, etc.; in the process, B received the SMS related to the i-PIN authentication sent by the item trading site. Table 4 shows the overall attack contents in chronological order.

V. DISCUSSION

Most of the existing malware detection and analysis studies only target app installation files. Therefore, it takes too much time to perform a static or dynamic analysis on all files or executable files present on the device.

We proposed a model of investigation in digital forensics to detect and analyze IoT malware quickly and effectively. Unlike conventional methods, our model makes it possible to identify suspected malware without analyzing all executable files by analyzing file system metadata, operating system logs, application logs, etc. This will be useful for hacking incidents that require quick incident response and recovery.

It is also essential to determine when and how malware broke into the device in a hacking incident investigation. Considering the characteristics of malware, our model can analyze the timing and method of penetration of skillful malware using social engineering methods by comprehensively analyzing text messages, web browser logs, and the creation time of app installation files from a digital forensic perspective.

Besides, recent malware uses many droppers and uses clever methods to induce users to delete vaccines during the hacking process. We presented a response to how to analyze this malware through real-world case analysis.

Finally, hackers typically recycle existing code when developing malware. Therefore, we proposed a method of collecting information to track and group hacker groups through the similarity of logic, the hash value of essential libraries, information of executable files, information of C&C server, etc. through executable reverse engineering.

Our findings are expected to contribute to the efficient detection and analysis of malware applied with social engineering techniques – which have been occurring a lot in IoT devices and becoming a big issue nowadays – and to the detailed analysis of malicious behavior.

VI. CONCLUSION AND FUTURE WORK

Personal information such as call history, message records, web browser usage records, photos, financial information, and business-related information is stored in IoT devices.

Future wearable devices are also expected to store biometric information such as blood pressure, heart rate, and electrocardiogram. As these IoT devices increase, the number of malware stealing various kinds of information inside the devices is also increasing. Thus, there is an increasing need for a study on IoT devices' incident response from a digital forensic perspective.

We analyzed IoT devices equipped with Android, iOS, and Linux OS software involved in actual infringement accidents due to malware combined with social engineering techniques. As a result, we found out when and how this clever malware broke into devices and what malicious behavior it did and analyzed information that could track down hackers.

Furthermore, through the analysis results, we have created a digital forensics investigation model for effective incident response in the event of an infringement accident caused by malware on IoT devices. This model was developed to detect malware quickly by screening and analyzing artifacts that must be analyzed through malware characteristics among multiple files of IoT devices.

We focused on Android because many IoT devices now adopt the Android OS software. As more various IoT devices are expected to be released in the future, research on malware detection will also be needed on IoT devices with more diverse OS software such as Tizen, WebOS, and ROS.

AUTHOR CONTRIBUTIONS

Investigation, Dohyun Kim; Writing, Dohyun Kim and Jong Hyuk Park (review & editing), Dohyun Kim (original draft) and Yi Pan (editing); Methodology, Dohyun Kim; Validation, Dohyun Kim and Yi Pan; Resources, Dohyun Kim; Visualization, Dohyun Kim; Formal analysis, Dohyun Kim; Supervision, Jong Hyuk Park; Project administration, Dohyun Kim; Funding acquisition, Yi Pan and Jong Hyuk Park.

REFERENCES

[1] Cisco. *Cisco Visual Networking Index: Forecast and Trends 2017-2022*. Accessed: Sep. 2020. [Online]. Available: https://networking.report/Resources/Whitepapers/08d1985b-0247-42f2-ab7f-2c3e6e322821_white-paper-c11-741490.pdf/

[2] V. Chebyshev. *Mobile Malware Evolution 2019*. Kaspersky. Accessed: Sep. 2020. [Online]. Available: https://cyrekdigital.com/pl/blog/content-marketing-trendy-na-rok-2019/white-paper-c11-741490.pdf/

[3] R. Samani. *McAfee Mobile Threat Report Mobile Malware Is Playing Hide and Steal*. McAfee. Accessed: Sep. 2020. [Online]. Available: https://www.mcafee.com/content/dam/consumer/en-us/docs/2020-Mobile-Threat-Report.pdf/

[4] M. Ghasemi, M. Saadaat, and O. Ghollasi, "Threats of social engineering attacks against security of Internet of Things (IoT)," *Fundam. Res. Electr. Eng.*, pp. 957–968, 2019.

[5] Statcounter. *Mobile Operating System Market Share Worldwide*. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/

[6] E. O. Yeboah-Boateng, E. Osei, and P. M. Amanor, "Phishing, SMiShing & Vishing: An assessment of threats against mobile devices," *J. Emerg. Trends Comput. Inf. Sci.*, vol. 5, no. 1, pp. 297–307, 2014.

[7] M. Foozy, C. Feresa, R. Ahmad, and M. F. Abdollah, "Phishing detection taxonomy for mobile device," *Int. J. Comput. Sci. Issues*, vol. 10, no. 1, pp. 338–344, 2019.

[8] H. Shahriar, T. Klintic, and V. Clincy, "Mobile phishing attacks and mitigation techniques," *J. Inf. Secur.*, vol. 6, no. 3, pp. 206–212, 2015.

[9] N. Choudhary and A. K. Jain, "Comparative analysis of mobile phishing detection and prevention approaches," in *Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst.*, May 2017, pp. 349–356.

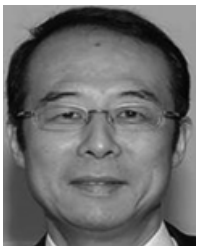
[10] A. Qamar, A. Karim, and V. Chang, "Mobile malware attacks: Review, taxonomy & future directions," *Future Gener. Comput. Syst.*, vol. 97, pp. 887–909, Aug. 2019.

[11] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," *Future Internet*, vol. 11, no. 4, p. 89, Apr. 2019.

[12] D. Goel and A. K. Jain, "Smishing-classifier: A novel framework for detection of smishing attack in mobile environment," in *Proc. Int. Conf. Next Gener. Comput. Technol.*, Singapore, Oct. 2017, pp. 502–512.

[13] J. W. Joo, S. Y. Moon, S. Singh, and J. H. Park, "S-detector: An enhanced security model for detecting Smishing attack for mobile computing," *Telecommun. Syst.*, vol. 66, no. 1, pp. 29–38, Sep. 2017.
[14] A. K. Jain and B. B. Gupta, "Rule-based framework for detection of Smishing messages in mobile environment," *Procedia Comput. Sci.*, vol. 125, pp. 617–623, 2018.
[15] S. Wang, Z. Chen, Q. Yan, B. Yang, L. Peng, and Z. Jia, "A mobile malware detection method using behavior features in network traffic," *J. Netw. Comput. Appl.*, vol. 133, pp. 15–25, May 2019.
[16] M. Alazab, M. Alazab, A. Shalaginov, A. Mesleh, and A. Awajan, "Intelligent mobile malware detection using permission requests and API calls," *Future Gener. Comput. Syst.*, vol. 107, pp. 509–521, Jun. 2020.
[17] Z. Xu, C. Shi, C. C.-C. Cheng, N. Z. Gong, and Y. Guan, "A dynamic taint analysis tool for Android app forensics," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 160–169.
[18] M. Sun, T. Wei, and J. C. S. Lui, "TaintART: A practical multi-level information-flow tracking system for Android RunTime," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 331–342.
[19] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. Comput. Syst.*, vol. 32, no. 2, pp. 1–29, Jun. 2014.
[20] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps," *ACM SIGPLAN Notices*, vol. 49, no. 6, pp. 259–269, Jun. 2014.
[21] L. Li, A. Bartel, T. F. Bissyande, J. Klein, Y. Le Traon, S. Arzt, S. Rasthofer, E. Bodden, D. Octeau, and P. McDaniel, "IccTA: Detecting inter-component privacy leaks in Android apps," in *Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng.*, vol. 1, May 2015, pp. 280–291.
[22] M. I. Gordon, D. Kim, J. H. Perkins, L. Gilham, N. Nguyen, and M. C. Rinard, "Information flow analysis of Android applications in droidsafe," in *Proc. NDSS*, Feb. 2015, vol. 15, no. 201, p. 110.
[23] F. Wei, S. Roy, and X. Ou, "Amandroid: A precise and general inter-component data flow analysis framework for security vetting of Android apps," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1329–1341.
[24] L. Qiu, Y. Wang, and J. Rubin, "Analyzing the analyzers: FlowDroid/IccTA, AmanDroid, and DroidSafe," in *Proc. 27th ACM SIGSOFT Int. Symp. Softw. Test. Anal. (ISSTA)*, Jul. 2018, pp. 176–186.
[25] A. Kumar, V. Agarwal, S. Kumar Shandilya, A. Shalaginov, S. Upadhyay, and B. Yadav, "PACER: Platform for Android malware classification, performance evaluation and threat reporting," *Future Internet*, vol. 12, no. 4, p. 66, Apr. 2020.
[26] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "EMULATOR vs REAL PHONE: Android malware detection using machine learning," in *Proc. 3rd ACM Int. Workshop Secur. PrivacyAnalytics (IWSPA)*, Mar. 2017, pp. 65–72.
[27] A. Nieto and R. Rios, "Cybersecurity profiles based on human-centric IoT devices," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 39, Dec. 2019.
[28] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, p. 3, Dec. 2018.
[29] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for Machine-Learning-Based Android malware detection," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3216–3225, Jul. 2018.
[30] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning," *Tsinghua Sci. Technol.*, vol. 21, no. 1, pp. 114–123, Feb. 2016.
[31] G. Suciu, C.-I. Istrate, R. I. Răducanu, M.-C. Diţu, O. Fratu, and A. Vulpe, "Mobile devices forensic platform for malware detection," in *Proc. Int. Symp. ICS SCADA Cyber Secur. Res*, vol. 6, Sep. 2019, pp. 59–66.
[32] K. Yau, K. P. Chow, and S. M. Yiu, "An incident response model for industrial control system forensics based on historical events," in *Proc. Int. Conf. Critical Infrastruct. Protection*, Mar. 2019, pp. 311–328.
[33] A. Dimitriadis, N. Ivezic, B. Kulvatunyou, and I. Mavridis, "D4I–Digital forensics framework for reviewing and investigating cyber attacks," *Array*, vol. 5, Mar. 2020, Art. no. 100015.
[34] K. A. Torkura, M. I. H. Sukmana, F. Cheng, and C. Meinel, "SlingShot-automated threat detection and incident response in multi cloud storage systems," in *Proc. IEEE 18th Int. Symp. Netw. Comput. Appl. (NCA)*, Sep. 2019, pp. 1–5.
[35] J. Thomas, R. P. Galligher, M. L. Thomas, and G. Galligher, "Enterprise cybersecurity: Investigating and detecting ransomware infections using digital forensic techniques," in *Enterprise Cybersecurity: Investigating and Detecting Ransomware Infections Using Digital Forensic Techniques* (Computer and Information Science), vol. 12, no. 3, J. E. Thomas, R. P. Galligher, M. L. Thomas, and G. C. Gallilgher, Eds. Richmond Hill, ON, Canada: Canadian Center of Science and Education, 2019, pp. 72–80. [Online]. Available: http://www.ccsenet.org/journal/index.php/cis
[36] *App Download and Usage Statistics*. Accessed: Sep. 2020. [Online]. Available: https://www.businessofapps.com/data/app-statistics/
[37] *JEB Decompiler*. Accessed: Sep. 2020. [Online]. Available: https://www.pnfsoftware.com/jeb/
[38] *IDA Pro*. Accessed: Sep. 2020. [Online]. Available: https://www.hex-rays.com/products/ida/
[39] *Taig Jailbreak*. Accessed: Sep. 2020. [Online]. Available: https://pangu8.com/taig/
[40] *Clutch*. Accessed: Sep. 2020. [Online]. Available: https://github.com/KJCracks/Clutch/
[41] *iTools*. Accessed: Sep. 2020. [Online]. Available: https://www.itools4.com/
[42] *AhnLab*. Accessed: Sep. 2020. [Online]. Available: https://www.ahnlab.com/
[43] *Binwalk*. Accessed: Sep. 2020. [Online]. Available: https://github.com/ReFirmLabs/binwalk/

DOHYUN KIM received the Ph.D. degree from the School of Cybersecurity, Korea University, Seoul, South Korea, in 2019. He is currently an Assistant Professor with the Department of Computer Science, Catholic University of Pusan (CUP), Busan, South Korea. From September 2019 to March 2020, he was a Research Professor with the Digital Forensic Research Center (DFRC), Institute of Cyber Security and Privacy (ICSP), Korea University. From July 2017 to August 2019, he was a Researcher with the Intelligent Convergence Research Laboratory, Cyber Security Research Division, Vulnerability Analysis Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. He is also a Managing Editor of Human-centric Computing and Information Sciences (HCIS) (Springer) and an Associate Editor of the *The Journal of Information Processing Systems* (JIPS) (KIPS), and the *Journal of Digital Forensics* (Korea Digital Forensic Society (KDFS)). His research interests include digital forensics, cyber security, and vulnerability research.



YI PAN (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from Tsinghua University, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science from the University of Pittsburgh, Pittsburgh, in 1991. He is the Chair and a Professor with the Department of Computer Science and a Professor with the Department of Computer Information Systems, Georgia State University, Atlanta. His research interests include parallel and distributed computing, networks, and bioinformatics. He has published more than 100 journal articles with 38 articles published in various IEEE journals. In addition, he has published more than 100 papers in refereed conferences. He has also authored/edited 34 books (including proceedings) and contributed many book chapters. He has organized several international conferences and workshops and has also served as a program committee member for several major international conferences, such as BIBE, BIBM, ISBRA, INFOCOM, GLOBECOM, ICC, IPDPS, and ICPP. He has delivered more than ten keynote speeches at many international conferences and is a speaker for several distinguished speaker series. He is listed in Men of Achievement, Who's Who in Midwest, Who's Who in America, Who's Who in American Education, Who's Who in Computational Science and Engineering, and Who's Who of Asian Americans. He has served as the editor-in-chief or an editorial board member of 15 journals, including six IEEE TRANSACTIONS, and a guest editor for ten journals, including the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS and the IEEE TRANSACTIONS ON NANOBIOSCIENCE.

JONG HYUK (JAMES) PARK (Member, IEEE) received the Ph.D. degrees from the Graduate School of Information Security, Korea University, South Korea, and from the Graduate School of Human Sciences, Waseda University, Japan. From December 2002 to July 2007, he had been a Research Scientist of Research and Development Institute, Hanwha S&C Company Ltd., South Korea. From September 2007 to August 2009, he had been a Professor at the Department of Computer Science and Engineering, Kyungnam University, South Korea. He is currently a Professor with the Department of Computer Science and Engineering and the Department of Interdisciplinary Bio IT Materials, Seoul National University of Science and Technology (SeoulTech), South Korea. He has published about 200 research papers in international journals and conferences. His research interests include the IoT, human-centric ubiquitous computing, information security, digital forensics, vehicular cloud computing, and multimedia computing. He is a member of the IEEE Computer Society, KIPS, and KMMS. He received the best paper awards from ISA-08 and ITCS-11 conferences and the outstanding leadership awards from IEEE HPCC-09, ICA3PP-10, IEE ISPA-11, PDCAT-11, and IEEE AINA-15. He also received the outstanding research awards from SeoulTech, in 2014. He has been serving as the chair, program committee, and organizing committee chair for many international conferences and workshops. He is the steering chair of international conferences—MUE, FutureTech, CSA, CUTE, UCAWSN, and World IT Congress-Jeju. He is the Editor-in-Chief of *Human-centric Computing and Information Sciences* (HCIS) (Springer), *The Journal of Information Processing Systems* (JIPS) (KIPS), and *Journal of Convergence* (JoC) (KIPS CSWRG). He is an associate editor/editor of 14 international journals, including JoS, JNCA, SCN, and CJ. In addition, he has been serving as a guest editor for international journals by some publishers: Springer, Elsevier, Wiley, Oxford University Press, Emerald, Inderscience, and MDPI.