국제학석사 학위논문

A Study on Internet of Things (IoT) Forensic

-Smart Home IoT Data Acquisition Procedure-

사물인터넷 포렌식에 관한 연구

-스마트홈 IoT 데이터 획득 절차-

장수봉 (Jang, Subong)

국제학과 (International Studies)

정보법과학전공 (Legal Informatics and Forensic Science)

한림대학교 대학원

Graduate School, Hallym University

국제학석사학위논문

A Study on Internet of Things(IoT) Forensic

2019

장수봉

국제학석사 학위논문

A Study on Internet of Things (IoT) Forensic

-Smart Home IoT Data Acquisition Procedure-

사물인터넷 포렌식에 관한 연구

-스마트홈 IoT 데이터 획득 절차-

장수봉 (Jang, Subong)

국제학과 (International Studies)

정보법과학전공 (Legal Informatics and Forensic Science)

한림대학교 대학원

Graduate School, Hallym University

Joshua I. James, 장윤식 교수지도

국제학석사 학위논문

장수봉의 석사학위논문을 합격으로 판정함

2019년 6월 28일

심사위원장 박노섭

_____

심사위원 안정민

_____

심사위원 장윤식

_____

심사위원 Joshua I. James

_____

Table of Contents

List of Figures

List of Tables

CHAPTER 1. INTRODUCTION

It is now possible to monitor and remotely control household appliances in the house such as television, refrigerator, temperature, door, light, etc. via the Internet. Furthermore, the generated data from the user's activities can be used to analyze user can provide convenient services to the user. For example, a coffee pot automatically makes coffee the wakes-up or the boiler can warm up the temperature of the house before we get home. These are called Internet of things (IoT). The term Internet of thing was first mentioned by Kevin Ashton in 1999. However, this idea has existed since the 1970s and was called "the embedded internet", "pervasive internet" and others.

There is no official definition of the internet of things. IEEE described the Internet of Things as: "A network of items – each embedded with sensors – which are connected to the internet [1]". Internet Society (ISOC) described the Internet of Things as: "The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate, exchange and consume data with the minimal human intervention[2]". In addition, many organizations and groups are defining Internet of things (IoT). However, most definitions indicate the expansion of Internet connectivity through embedded and sensor networks. So, if an object provides services to humans via the Internet, it can be called Internet of Things (IoT).

As technology evolved, many technologies and the infrastructure were connected, the development of the IoT increased exponentially. IDC, a US market research and consulting firm for IT and telecommunications, consumer technology, forecasts that a total of 150 billion things will be connected and will produce 175zetabyte of data. And average people will have nearly 5,000 digital interactions per day by 2025 [3]. IoT technology is applied to various fields such as

smart home, wearable, smart city, industry, automobile, transportation, health and provides convenient and useful programs[4]. As the dependency of data increases, so does the potential for digital evidence.

In the IoT ecosystem, potential data from crime scenes can be expanded because the data generated by sensors through human intervention is vast and criminals can not cognize all of the data generated. Unlike traditional forensics, which has found data evidence on mobile, PC, CCTV and limited sources, data sources can be increased and data evidence can be more critical because of the IoT Characteristic that any object can be Internet of Things (IoT). But the vast amount of data is a challenge for digital forensics investigators to identify accurate evidence. IoT forensic investigations have many challenges due to the incompatibility of digital forensic tools and standard forensic methodologies currently available in the IoT environment[5]–[7].

1.1. Internet of Things (IoT) Forensic

With the increasing use of IoT technology, the need for IoT Forensic is increasing. Unlike traditional digital forensics, IoT forensics has become a complex form of forensics due to the unique forms of devices, operating systems, and services that each company has. For example, in traditional forensics, the source of the evidence could be mobile, cell phone, server, gateway, and the source of the IoT forensic evidence could be home appliances, TV, medical implant in human or animal, tag reader, sensor node, tag reader. In addition, IoT technology can be different in jurisdiction and ownership because it also uses the cloud to share and deliver services with a wide range of companies in a wide range of regions. [8].

IoT is a combination of different areas. And It can be divided into cloud, network, and device areas. [6]–[8]. Even if an investigator finds a device that could be evidence at a crime scene, the data could be stored in a cloud server located in another jurisdiction. And because the cloud uses

virtual computing, data can be erased if there is a server reboot or synchronization problem with the client device. In addition, IoT forensics has many challenges. The IoT forensic challenge is discussed in Chapter 2. So far, there have been some real cases of IoT forensics. Examples include Fitbit, Amazon Alexa, and Pacemaker, which help solve challenges.

1.2. Significant of the study

Important of digital evidence from IoT - Because IoT's data is less likely to be modulated and is more reliable, it can be potential electronic evidence. The reason is next. IoT provides services with minimal human intervention. It does not inject data directly by a human. It generates data and provides services based on the activity, temperature, time, location, etc. of the human or object according to its settings and purpose. And the generated data is stored somewhere like a cloud or a small storage space for learning purposes. As a result, the IoT environment may contain contextual evidence that the assailant simply ignores. In order to maliciously modulate the data, it is necessary to know the IoT device in advance and access the storage space using a network and a physical method. Since each device has a different form, it is not easy for the general user to access the data.

Practical IoT investigation procedure model - While there is a lot of data to process and expertise is required of investigators to conduct IoT investigations, there is a lack of tools and methodologies that can be investigated efficiently. Many studies have presented tools and methodologies, but there are many that are not practical or not realistic[5]. Since many of the studies carried out were too specific or too general without experiments, it was difficult to use them as a practical IoT investigation procedure model[9]. However, partial step-by-step testing is possible. We discuss and generalize methods that can be tried on a cloud, network, device area in the acquisition step so that investigators can try to acquire it directly, even if there is no known tool or method for acquisition.

1.3. Scope and limitation

Developing and complementing the IoT investigation procedure model is one of IoT Forensic's challenges due to unavailable test or/and environment. This paper was written to improve the IoT investigation procedure model for house using research perspective. We analyzed four Korean speakers, two US foreign speakers, two smart home kits, a thermostat, and raspberry pie. And the study was divided into the areas of cloud, client app, network, and physical to find acquisition method, the data location, Types of data we can get and so on from the IoT devices for smart house.

To ensure that IoT evidence can be accepted in court, the existing general digital forensic procedures that have already been verified and trusted should be improved to fit the IoT. Many researchers have worked for the IoT investigation procedure model, and they have defined the model in several steps. However, there were many challenges to building a test environment, the whole model could not be tested. Also, a small number of studies were highly specialized and could not generalize the IoT investigation procedure model[9]. Therefore, the whole IoT investigation procedure model will be completed when the test is performed for each step. This paper will propose a tested practical IoT Data Acquisition procedure for the Acquisition step in the entire IoT Investigation procedure.

There are limitations to this paper.

- The purpose of this experiment(case study) is to apply real IoT devices to the IoT Data Acquisition procedure presented. However, there is a limit to experimenting with all of them because many IoT devices in various forms exist in the IoT world.
- The procedure proposed in this paper is for acquisition. A simple analysis to obtain credential information such as ID, password, token and cookie in the IoT

ecosystem or to check what type of data exists in client data or device data for next-best-thing triage can do. However, data analysis will not be covered in detail.

1.4. Thesis structure

This thesis is organized as follows: Chapter 2 presents the review of previous works in the field of IoT Digital Forensics - IoT Forensic Challenges, existing IoT acquisition methods, and previously proposed IoT Investigation procedure models. Chapter 3 presents our proposed IoT acquisition procedure based on the research experiences we conducted on the IoT devices. In Chapter 4 we presented the practical tests for 5 IoT devices using the proposed IoT acquisition procedure. Finally, in Chapter 5 we presented the discussions on the results, conclusions and possible future research directions to the proposed model.

CHAPTER 2. BACKGROUND RESEARCH

This chapter describes existing research for IoT digital forensics. The first section will review the challenges of IoT forensics, and the second section will review the acquisition method. And the third section will review the already presented IoT Investigation model.

2.1 The challenge of IoT forensic

IoT is created by gathering various areas. This area includes mobile, computers, sensors, various network devices, the cloud, VR, and AI. In other words, the IoT Forensic Challenge includes the Forensic Challenge for each area [7].

- Visibility - The main challenge for the IoT crime scene is visibility. IoT device may be invisible or numerous and may not be identified until the device is connected to the network, or even the device may be inserted inside the body. In other words, it may not be possible to identify all IoT devices[7], [10].

- Difficulty in memory dump – Dumping data at the scene during acquisition is a challenge. There are a lot of things to consider because dumps can be difficult or impossible with existing tools. And the data format, the data location, acquisition method, etc may be different for each device [7], [10], [11]. In IoT environment, data is mostly stored and processed in the cloud. However, obtaining access to data for investigation purposes is not appropriate for IoT investigators due to IoT's service contract restrictions[6], [12].

- Big IoT data analysis – Analyzing the data and finding information related to the crime can help with the investigation. But increasing the amount of data and increasing complexity hinders the investigation. Because IoT data is a mixture of both structured and unstructured data, Effective tools are needed to deal with this[6], [12].

- Volatile (easy to change) – Volatile memory contains useful information and should not be missed when possible [10]. IoT devices such as sensors, however, use small memory for efficient energy consumption, and when data is generated, it overwrites the memory immediately due to small storage space. In other words, crime scene data may be lost if an investigator accesses the device[13], [14].

- Jurisdiction – In IoT environment, user data is stored in multiple locations. The location may be another jurisdiction, such as a country or a region. That's why if data from a device found at a crime scene requested to vender, it may take too long to received or rejected[6], [7], [12].

2.2 IoT data acquisition method

The IoT data Acquisition method performed is divided into three major areas: Cloud Acquisition, Network Acquisition, Device Acquisition. In this section, we will review the research performed for each acquisition method.

2.2.1 Cloud acquisition

The cloud is a shared collection of network resources (e.g., networks, servers, storage, applications and services) that can serve a wide area. That is, it is a convenient computing service where users can receive data sharing and service delivery regardless of physical location. And because the cloud is a subgroup of network services, cloud forensics is also a subgroup of network forensics [15].

An initial study of cloud forensics was conducted on the client side. Client-based cloud forensic is the collection and analysis of data stored on the client side such as mobile apps or browser apps in PC that communicates with the cloud. As a result, the possibility

of forensics such as Google Docs, Amazon S3, Dropbox, Evernote, ownCloud, etc. has been confirmed[16]–[18]. Since then, research has been conducted to acquire data from the cloud itself using the API. This method points out that the client device does not persist data and is the method to overcome that limitation. As a result, the possibility of forensics such as Google Drive, Microsoft OneDrive, Dropbox has been confirmed[19].

H. Chung et al. in [14] proposed a forensic approach to Amazon Alexa, which combines cloud-side and client-side forensics. They accessed and collected cloud-native artifacts using unofficial APIs and developed the Amazon Alexa forensic tool kit in that way. This approach has two limitations: that user information (ID, PW) is required and that deleted data is difficult to recover. This means that if the investigator cannot obtain user information (ID, PW), or if the user has deleted cloud data, it will be difficult to get data from the Amazon Alexa cloud. To obtain unofficial APIs, they analyzed traffic using the tool that is Charles web debugging proxy. Unofficial APIs can be identified in the returned value on the condition that they are valid user credentials.

2.2.2 Network acquisition

IoT is a diverse set of technologies and communicates over various communication protocols such as LTE, Wi-Fi, Bluetooth, Zigbee, and Z-wave depending on geographical location. And IoT appliance integrates cloud application through API. Packets can be traced using network equipment such as routers, switches and Software Defined Networking (SDN) switches to collect and analyze these communications or to use packet-tracking tools such as Wireshark [6].

Using packet analysis tools such as Wireshark is the basis of network forensics. Wireshark can collect and analyze packets and is a great tool for discovering a wide

range of security threats and attacks. [20]. If the packet is not encrypted, it is easier to check the data via Wireshark, but since many devices have recently encrypted the data, it is difficult to see the data in the packet. H. Chung et al. in [14] used Charles web debugging to verify encrypted traffic and found Amazon Alexa's unofficial APIs in the returned value. And they were able to collect and analyze cloud data using informal APIs. While this method is not described in detail, it will help you to find unofficial APIs for other devices using similar IoT ecosystems.

2.2.3 Device level acquisition

Device Level Acquisition is the acquisition of data within a physical device[6]. JTAG, Serial port(debug pad) and Chip-off are representative. However, this method is not easy and requires a great deal of attention because it has a high possibility of damaging the device. Processes can be very difficult, such as soldering, special tools, or connecting devices and computers, but if the connection is successful, the reading process is easy and the dump is the same as any other physical acquisition [21].

Clinton et al. [22] performed a device level acquisition of Amazon Alexa. The team disassembled Amazon Alexa to figure out its internal structure and found a suspicious JTAG port on the board. They did not attempt to acquire data via JTAG in the absence of special tools, but they said it was this is a promising avenue that could allow full control of the device if successful." They understood the boot process and checked the UART on the Debug Pad at the bottom of Amazon Alexa and succeeded data dump. But This method has been disabled since 2017 due to a patch to compensate for physical vulnerabilities in Amazon[23].

2.3 Presented IoT investigation models

This section reviews the models presented for IoT Digital forensic. These models were presented for all the processes from the time of the incident through the investigation to the submission to the court.

Oriwoh et al. in [7] proposed a model that consists of three zones and Next-Best-Thing Triage (NBT) Model for use in conjunction with the three zones. These zones help guide investigators where to begin their work with regards to an investigation. Zone 1 is made up of the internal zone which includes all the hardware, software, and networks that resides in the actual crime scene. IoT devices such as temperature control, door sensors, smart cameras, and Hubs reside here and could be helpful in the IoT forensics. Zone 2 consists of all devices and software on the edges of the network that provide a communication medium between the internal and external networks. Zone 3 consists of any hardware or software outside of the internal network. This could include cloud, social network, or Internet Service Providers (ISP). This zonal model in approaching the investigation can be useful to plan and identify possible digital evidence from IoT devices. They also proposed 4 phases for IoT forensics methodology; Preparation which is identifying any possible evidence, Acquisition which is Imaging and retrieving data, Investigation which is the actual criminal investigation, and Reporting and storage. In the phase of Investigation, some sources of evidence might become unavailable after a crime is committed; e.g. a mobile phone might be disposed of. In this case, the Next-Best-Thing model can be applied. Identify what devices were connected to the Object-Of-Forensic-Interest (OOFI) and what slice of evidence, if anything, are left behind after its removal from the network. The proposed work can provide many benefits such as effective and efficient IoT-related investigation in terms of identifying relevant evidence. However, developing and testing this work is challenging.

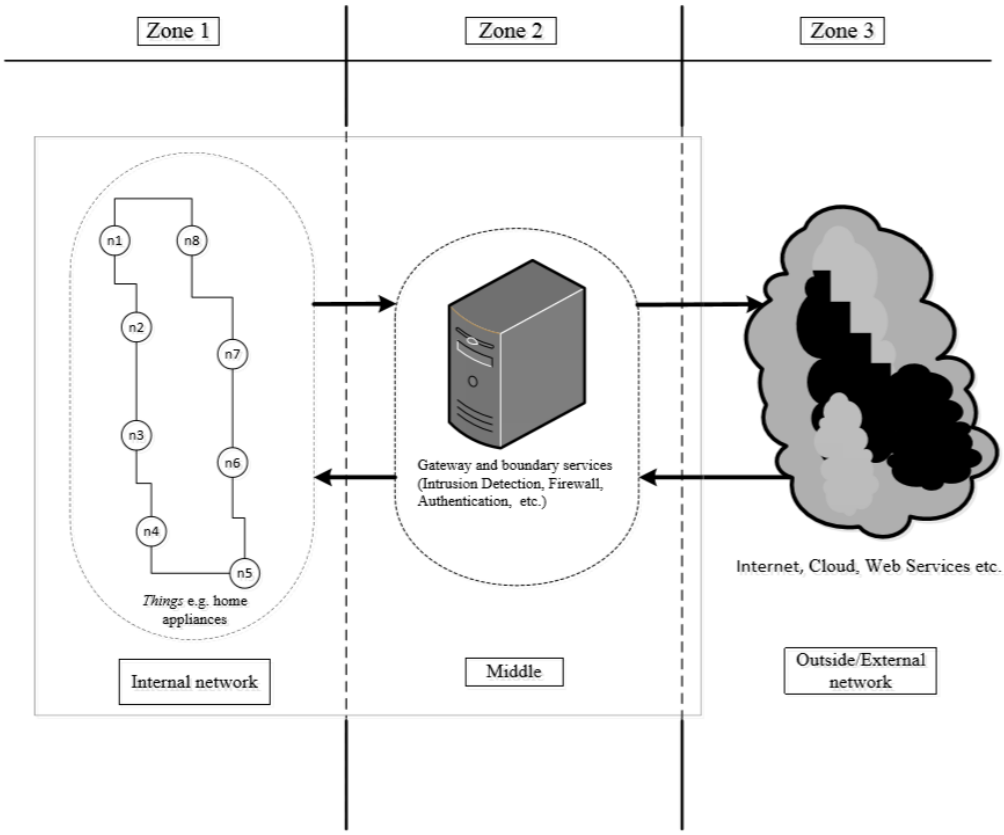

Figure 1: 1-2-3 Zones Model by Oriwoh et al., 2013

Perumal et al. in [5] aim to identify and design the best approach by designing a new model to follow while performing investigation situations for digital forensic professionals and law enforcement agencies. They proposed an integrated model based on triage model and 1-2-3 zone model. 1-2-3 zone model divides the crime scene territory into zones, Zone 1 work as the internal network where it resides all connection is based here. Zone 2 as the zone that resides all the device and the border network and last Zone 3 covers a huge data collection platform. This model starts with authorization, planning and obtaining a warrant – shares the same stages as the traditional forensic model. This paper's proposed model starts by identifying the base device, this means identifying machine to machine communication (M2M) or device to device communication. These communications would be Z-Wave, 4G, 3G, LTE, Wi-Fi, Ethernet, and Power Line Communication (PLC). The zone identification, described above, comes after communication

identification. The next stage of the model is the triage examination; the investigator should carefully investigate which data us fragile, structure data, or useful data. In this stage, the investigator encounters a collection of server clusters such as a router, gateway, Cloud platforms and Fog platforms. After the above stages, the procedure back to the traditional digital forensic investigation process which would be chain of custody, lab analysis, result, proof & defence and archive & storage. The proposed approach can help forensic experts conduct IoT investigations with a large size-based perspective.

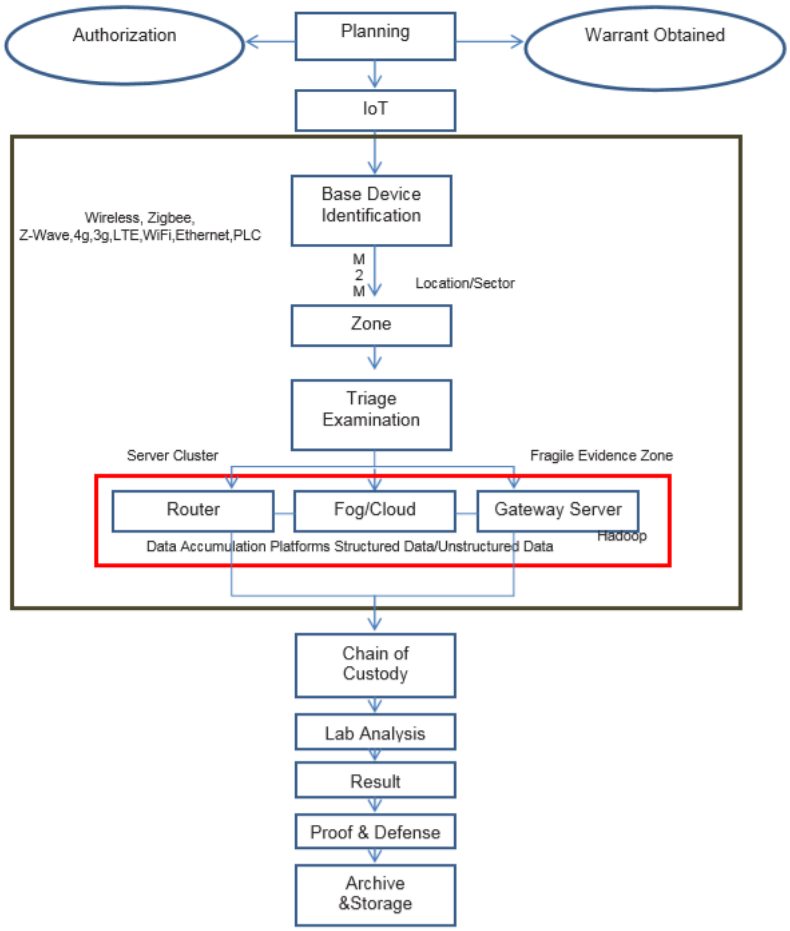

Figure 2: IoT Based Digital Forensic Model by Perumal et al., 2015

V. R. Kebande and I. Ray in [24] proposed a model based on model ISO/IEC 27043: 2015 that is international standards for information technology, security technologies, case investigation principles and processes. This model included ISO / IEC 27043: 2015 to ensure the

acceptability of digital evidence in court and included Digital Forensic Readiness (DFR) for IoT in preparation for potential security incidents in IoT environments. DFR consists of activities such as defining IoT scenarios, identifying IoT evidence sources, planning for incident detection, PDE collection, digital preservation, and potential evidence retention. The next step is the IoT Forensic procedure, which is divided into Cloud Forensic, Network Forensic, and Device Level Forensic. Reactive process procedures include initialization, collection processes, and investigation phases as the basis for digital forensic processes. And the entire process is concurrent with processes such as documentation, flow, and chain of custody to increase legal acceptability. In addition, this paper shows that it is more efficient by comparing with the existing models. This model is a holistic approach to IoT Digital Forensic and has the ability to integrate with other investigative processes, helping to complete the IoT Digital forensic in the future. However, it is not known whether it is practical because it has not been tested.

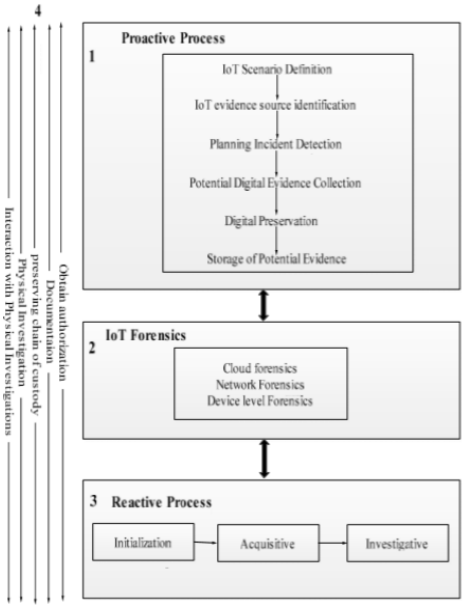

Figure 3: DFIF-IoT Framework by V. R. Kebande and I. Ray

CHAPTER 3. IOT DATA ACQUISITION PROCEDURE

This chapter presents the IoT acquisition procedure based on a research perspective. This chapter presents the IoT acquisition procedure based on a research perspective. This includes experience analyzing IoT devices for digital forensic tools.

3.1 Cloud acquisition

It is the process of acquiring aggregated user data, and specific setup and configuration data from the cloud if the IoT device uses the cloud as its backend. In fact, in the IoT ecosystem forensic investigation, the cloud should be prioritized as the major potential source of IoT evidential data. The data on the cloud might be preprocessed and available for user's consumption or in some structured format (JSON or XML format) for other services. Depending on the challenges faced due to the specific cloud setup, data acquisition can be performed either through automated tools or manually by browsing using web brothers. As a result, the specific data acquisition procedure depends on the challenges to be faced during the acquisition process. However, it is possible to put some generic best practices to be followed during the cloud data acquisition process.

General Procedures:

① Identify the cloud used as a backend for the IoT device to be investigated

② Get access to the cloud - ask the owners or the users to collaborate in accessing the cloud data or get accounts and credentials used to access the cloud. Since, almost all IoT backend clouds require account registration with user credentials (username and password), identifying the authentication methods and accounts used for the cloud is the critical step to acquired data from the cloud. Therefore, asking the

owners or administrative level users of the IoT device for the account and credentials used should be a priority in the process.

③ If the owners or users do not collaborate, that does not reveal the accounts and credentials, ask other family members including other potential users of the IoT within the crime scene scope/third persons such as relatives since some IoT devices can be shared for more than one person.

④ If access to the cloud through the user's collaboration is not successful, ask the cloud providers to collaborate to access user data using official judges letters/orders.

⑤ If all the above methods are not successful, look for unsecured APIs that does not need authentication to access user data. Some IoT devices may have security problems which may enable investigators to access user data.

⑥ If all the above methods do not reveal the accounts and credentials, forensically investigate the client smartphone applications such as companion apps used to monitor the IoT device through the cloud:

a. Look for username and password from the client apps file structure (shared preference files, SQLite dbs, caches). If usernames and passwords are ever available in the above files, mostly they might be encrypted. Therefore, in most cases decryption tools are also required for this process.

b. If there are no saved username and password, look for the cookies and tokens from the client apps. These cookies and tokens can be used to authenticate to the cloud and obtain user data from the cloud using APIs (official or unofficial).

b-1. Using official APIs provided by the product/device developer or cloud service providers

b-2. If the official APIs do not provide enough data or not successful at all, use unofficial APIs (APIs that are hidden but revealed by researchers)

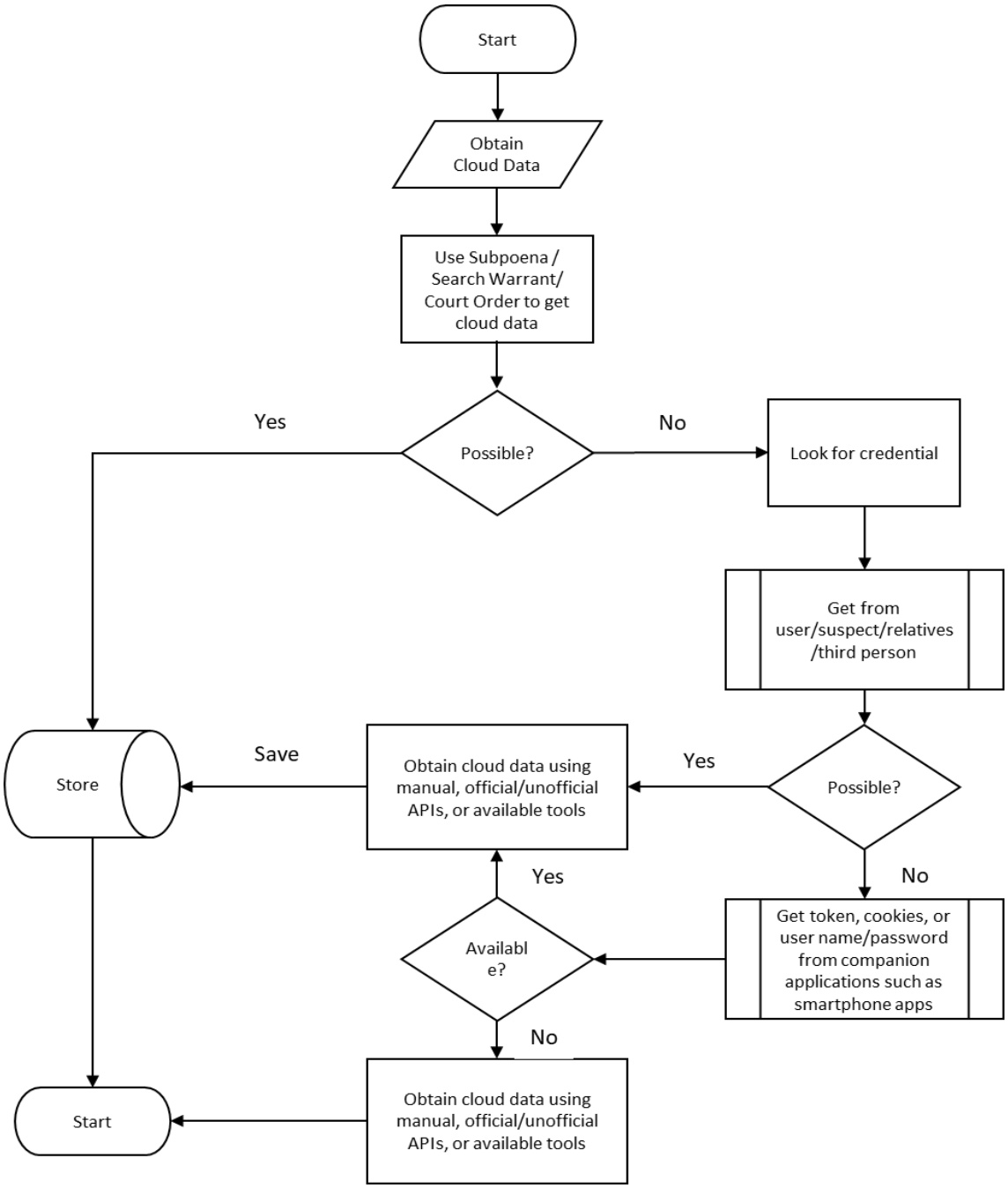

Figure 4: General methods for cloud acquisition

If the cloud forensics is not enough or do not reveal user data, obtain cache data or credentials from client applications such as smartphone and browsers used to access the cloud.

3.2 Client acquisition

Client apps are used by users to directly monitor, set up, and control IoT services. The client app has a browser app that uses the browser in PC or mobile, and the smartphone app that is built as a software program to directly controlled by a smartphone.

① Browsers and cache

In the IoT ecosystem, browsers are the other means to access the backend cloud used to aggregate and process IoT data. As such, when accessed to view or make changes to the IoTs configuration, browsers fetch and save some data on the local machine to facilitate local processing. This saved data can be extracted using forensic tools and procedures such as, browsers history examiner forensics tools[25]. The procedure for this browser depends on the available tool to perform the acquisition. Therefore, investigators need to decide the forensics procedures and tools to be used based on the browsers to be investigated including the machine the browser is running.

② Smartphone apps forensics

smartphone forensics can be used to collect user data, APIs, credentials, cookies from the:

a. IoT device configuring and monitoring app
b. IoT device related apps (integrated services apps) forensics

The smartphone forensics depends on the type of device and the available tools to be used to extract data. However, following already existing and matured smartphone forensics investigation procedures such as Guidelines on Mobile Device Forensics developed by NIST (NIST Special Publication 800-101

Revision 1) is helpful to perform apps forensic analysis [26].

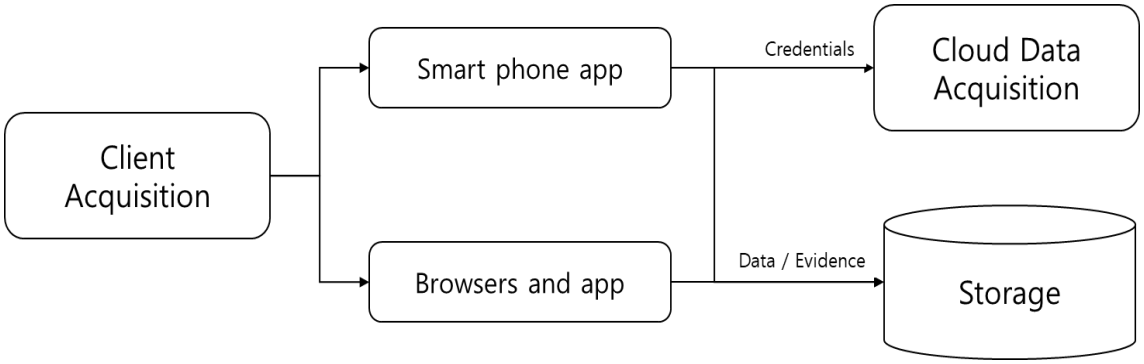

Figure 5: General methods for client acquisition

3.3 Network acquisition

Network analysis helps to capture raw data before it is structured and modified by the receiving ends to be further processed accordingly. Moreover, it helps investigators to identify the available devices and where they are communicating. Using the network analysis tools like Wireshark, Burp Suite and other procedures to intercept the traffic between the communicating objects, investigators can perform network forensics investigations. Similarly, IoT network communication forensics analysis can be achieved using different tools and procedures depending on the location of the intended interception and the required output.

① Web browsers network monitoring tool

This can be done using developer tools (DevTools) integrated into the browsers. This method is mostly used to identify the APIs and their structures for further data extraction from the cloud in manual or automated ways.

② Man-in-the-Middle attack tools

Man-in-the-middle attack tools can be used to intercept traffic between the communicating parties. This method can reveal important user data such as

credentials, live user commands and so on. The interception can be achieved at different locations based on the available tools, operation of the IoT and the required data.

a. IoT monitoring applications (Smartphone apps or browsers) and Cloud
   This process is intercepting the communication between cloud storage and the client applications used to access the IoT device and user data. This may reveal unencrypted user data and credentials, device settings and configurations, the communication protocol, the encryption technology (if used), and the APIs. In this scenario, interception tools such as Wireshark and Burp Suite can be used to intercept the traffic.
b. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device and the cloud may reveal unencrypted data traversing from the device to the cloud or from the cloud to the device or in both directions. Information such as device IDs, credentials, and user data, used protocol, and sensor signals with time stamps could be revealed by intercepting the communication between the IoT device/Hub and the cloud. It may also show, the communication protocol, using encryption and the authentication methods
c. Sensors and Device/Hub - intercepting the traffic between the sensors and device/hub may reveal both encrypted and unencrypted data - sensor IDs, credentials, and user data. Intercepting the traffic between device/hub and sensors depends on the RF communication type used between them. That means it may be Bluetooth, Z wave or Zigbee traffic. Other types of protocols, such as Bluetooth, Z wave or Zigbee, require a

dedicated interface module and can work with tools such as Wireshark. Because each protocol has a different data structure, it is necessary to analyze each protocol

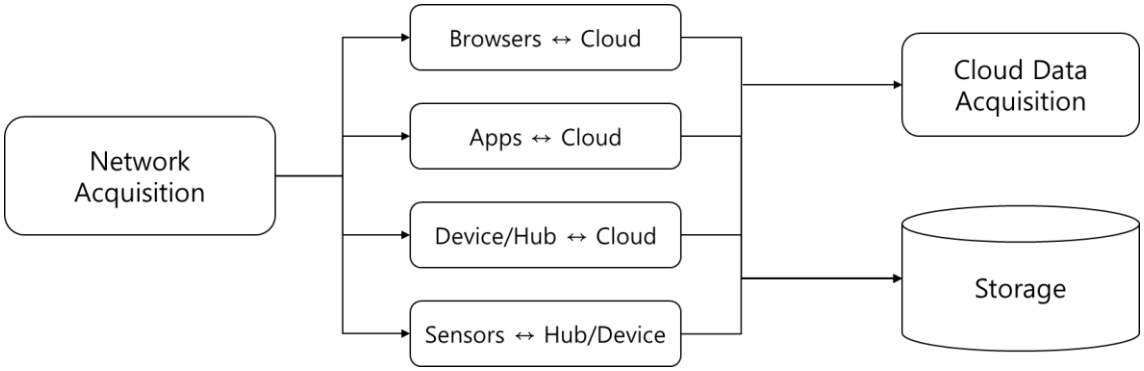

Figure 6: General methods for network acquisition

3.4 Device acquisition

Device level forensics includes the hardware forensics of the IoT/Hub device and its associated sensors. The device level forensics may reveal data [temporarily] saved or live data in memory, logs, device settings and unsynchronized data with the cloud or other connected devices. Since the device level forensics may include destructing the device and its operation, it should be considered as the least option if the other potential sources do not reveal the required data.

In order to extract data from IoTs using the hardware forensics options, there are challenges to be addressed and precautions need to be taken depending on the specific forensic procedures selected. In IoT device level forensics, the following challenges should be noted:

- Challenges to acquiring the data from these sensors and devices - the data changes dynamically even while investigators perform acquisitions, the limited size of memory, the heterogeneous form of the data, the time format and so on.

- Challenges to seizure and data acquisition process - due to the size of the object (e.g refrigerator) or the sensor (might be delicate), and operational principle (if disconnected may lose data)
- Challenges due to the application of the sensor/IoT device - the device might be used for critical application or service such as health monitoring
- Challenges in case of rooting and imaging - updates and fixes to the devices may be the main challenges to this method. This method is the less destructive option to perform hardware forensics on IoT devices. It requires to replace the operating system with a customized OS that can be rooted and controlled.

Data Acquisition on the device is likely to damage the device. Many IoT companies design or logically block devices so that there is no external interface for security. Therefore, you may need to disassemble the device for analysis. Since the device may not recover after disassembly, it should be checked from the way it is not damaged. Devices The procedure below has been presented in order not to damage the device.

① Using device-level APIs

Device level APIs may reveal locally saved data such as device logs, user data, configuration settings and so on. For instance, in Google OnHub, device-level APIs are available for developers to develop apps that interact with the device. The available APIs return JSON formatted data; and the device logs can be dumbed and analyzed. Available device level API identifiers are identified in the device vendor resources or in the open source information.

② Rooting and Imaging

Acquiring memory dump using imaging tools, Linux commands or booting

image over a provided interface such as USB. However, most recent IoT devices do not provide direct interfaces to access the device's memory, other acquiring methods should be considered. First, investigate whether the device can be in rooting, developer mode, or recovery mode. It examines whether the external appearance of the device has an interface that can interact with a device such as a USB. if the IoT device is switched to another mode, such as Routing, Developer Mode, or Recovery Mode, and interacts with the interface of the device's exterior, we can access the data of IoT device. In the case of the thermostat NEST, the user can control the device for other purposes via the developer mode change and the USB connection [27], [28].

③ Destructive method**s**

These methods are considered destructive because they need the disassembly of the device or removal of flash memory. These methods should be the last options because it is possible the board or other modules of the device could be damaged during the removal or the disassembly process.

1. JTAG - is a common hardware interface that provides the computer with a way to communicate directly with the chips on a board. It involves acquiring firmware data using standard Test Access Ports (TAPs). The data is transferred in a raw format. In the JTAG process, detailed training and expertise are required to identify and connect to the ports, creating customized boot loaders and recreating file systems.

2. Serial Port - is Universal Asynchronous Receiver/Transmitter, used for serial communications over a computer or peripheral device serial port such as UART, USB, I2P. Accessing the firmware via serial port and

extracting the data requires specialized interfaces and it is also an invasive technique which can reset the devices to factory settings resulting in loss of data. For example, Amazon Alexa has a JTAG suspect port on the board and a UART port at the bottom of the device, and some teams succeeded in extracting data through a UART port connection. Korean speaker SK NUGU has a USB port at the bottom and data can be extracted through the USB port. When disassembling other IoT devices, there may be a port inside the board or device. Depending on the number or markings of the PIN or PORT, we can predict which method will be used. However, recent devices often erase the markers or logically disconnect external physical ports[22].

3. Chip-off - refer to the acquisition of data directly from the device's flash memory. This extraction requires the physical removal of flash memory. Before doing the chip-Off extractions, investigators should consider the challenges they face such the device's variety of chip types, the raw data formats after extraction, and the risk of causing physical damage to the chip during the extraction process. For more information on chip-offs, refer to the "SWGDE Best Practices for Chip-Off" published by the Scientific Working Group on Digital Evidence (SWGDE) in 2016 [29].

The Destructive methods are already commonly used for forensics in mobile and embedded devices. Debug interfaces such as JTAG and UART are left for test and development prototypes after production. However, if it is exposed to an IoT device, the risk increases. in recent years it has become a patch to prevent physical access and can be difficult to access [23][27][30]. Although chip-off

may be more accurate to verify the data in memory, the result what we analyzed some IoT devices is that there are no data in some chip. Therefore, it is recommended to use this method as the last option.

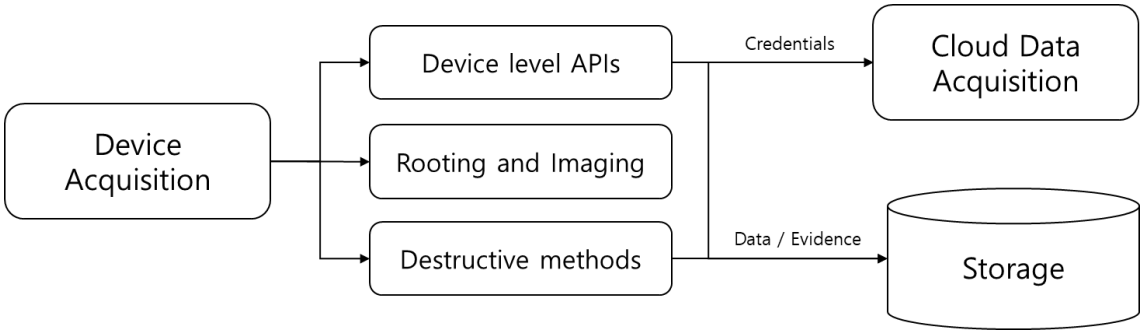

Figure 7: General methods for device acquisition

3.5 IoT data acquisition process

The exact location of IoT device data cannot be known unless preceded by prior research. Through the last several studies, we expect most of the data useful for the investigation to be in the cloud server. IoT device users use the client app on their PC and smartphone to control or monitor the services provided. And most of the user's data provided to the client app comes from the cloud. So, we expect the user's data to be stored in the cloud. However, cache or temporary data may remain or be stored on the user's smartphone or device. The data evidence required to resolve a criminal investigation varies from case to case. Therefore, if the type and location of the data are not known, investigators should consider the possibility of every side.

The IoT Data Acquisition process based on the proposed procedure is as Figure 8. If the data to be used as evidence is identified on each side of the Client and Device, the investigator can immediately acquire data from that side and move on to the analysis phase. If data is identified in the cloud, data from other networks, client, and device side is needed to obtain the credentials. Also, if an API is identified as an existing study, the API can be used, but if not, it can be an API on the network side. If the data in the cloud is identified, the investigator can use the credentials and API to access the cloud data. The credentials can be obtained from the Client or Device side, and the API can be obtained from a prior study or Network.

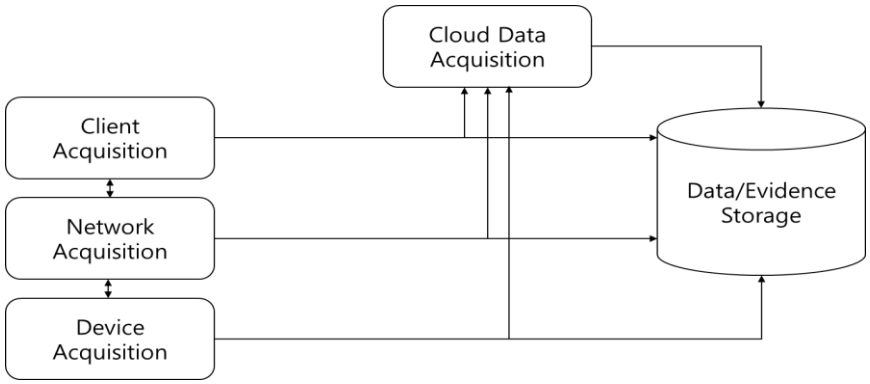

Figure 8: IoT Data Acquisition General Process

The IoT Data Acquisition Entire Architecture based on the proposed procedure is as Figure 9. As 1-2-3 ZONE model type by Oriwoh et al. in [7], it is divided according to network distance and includes methods which can be used on each side.

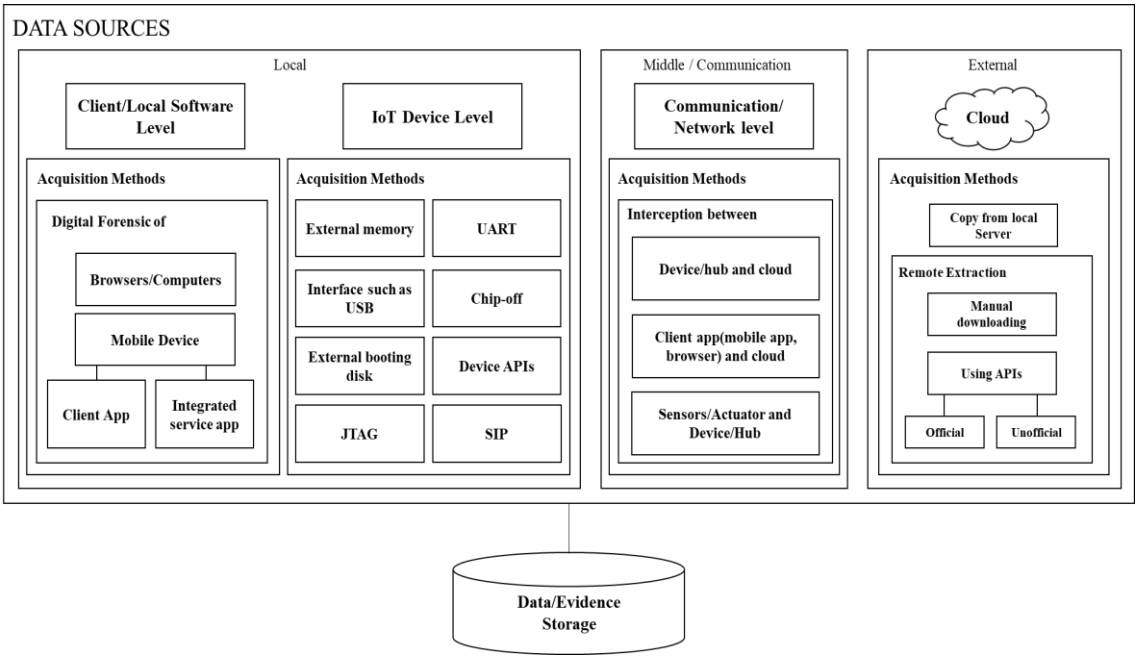

Figure 9: IoT Data Acquisition Entire Architecture

CHAPTER 4. CASE STUDIES

This chapter applies the actual IoT device according to the proposed IoT data acquisition procedure to test how much data can be obtained from the IoT device and whether the data is acquired. The actual IoT devices used in the tests are SK nugu, Naver Clova, Kakao mini, Giga genie, google home mini, which is an artificial intelligent speaker (AI speaker). AI speakers are expected to be owned by 40% of all households in Korea in 2019 and are connected to other IoT devices and have the ability to control the entire smart home by voice. In other words, there is a lot of direct involvement of users among IoT devices.[31]. We have set up a basic scenario before this case study. The reason for this is that the procedures are different depending on the situation. For example, if an investigator requests user data from a cloud provider and the cloud provider provides the data, the investigator does not need to look up APIs or credential information. Based on the basic scenario below, Cloud3 and Cloud4 will not be performed. AI speaker, smartphone and PC related to IoT ecosystem are confiscated.

The basic scenarios for following the IoT data acquisition procedure are as follows.

- The investigator identifies the IoT device found at the crime scene and performs the IoT data acquisition procedure.
- IoT device user has requested account (ID, PW), the user stated that the account information can not be memorized but it is logged in. And the user allowed to use all of the devices for investigation
- IOT device data was requested from the cloud provider, but the cloud provider rejected it for privacy reasons.
- The investigator has issued a warrant for the investigation and confiscated the IoT device, mobile, and PC.

The following are the tools and devices used in case studies.

- Extraction Tool – DD[1], MD-NEXT[2]
- Analysis Tool – MD-RED[3]
- Network Tool – Wireshark[4]

  SandroProxy[5]
- Smartphone - Galaxy Note 4(Android 6.0.1, 32GB)
- AI Speaker - Table 1

Table 1: Information of AI speaker used

NAME	Date of Manufacture	Model
SK Nugu	13th Jun, 2017	NU100WC
Naver Clova	Jan, 2018	NL-S110KR
Kakao mini	Jan, 2018	KM-1000
Giga Genie	Jun, 2017	KAO_CT11DD-SR
Google Home Mini	Dec, 2017	Home Mini

[1] **dd** is a command-line utility for Unix and Unix-like operating systems, the primary purpose of which is to convert and copy files.

[2] **MD-NEXT** is a commercial tool developed by Hancom GMD in Korea. It is a tool to extract evidence data stored in flash memory by connecting memory and main board for smartphone, tablet, etc.

[3] **MD-RED** is a commercial tool developed by Hancom GMD in Korea. It is a tool that analyzes extracted data images and restores deleted data.

[4] **Wireshark** is a free and open-source network packet analyzer.

[5] **Sandroproxy** is a mobile app tool for Man-in-the-Middle. it can capture, intercept, analyze, modify, replay http requests, web sockets

4.1 SK NUGU

SK Nugu is an Artificial Intelligence secretary developed by SK Telecom, Korea's major telecom company. and It is the first AI speaker that was launched in Korea in September 2016. And It provides services such as music/audio, phone, search, shopping/order, kids, finance, IoT device control, alarm, schedule, mood light, etc. In order to use SK Nugu, initial settings are required, and initial settings can only be made in conjunction with a smartphone. And since these services are provided in partnership with other vendors, sign in and login are required for each service in order to use other services. Once logged in, if the user does not intentionally log out, the application will continue to log in. SK Nugu account is linked with SKTelecom account.

4.1.1 SK Nugu Setting

In order to perform a case study, the actual IoT device must have data. We initialized it with the following account and generated the data with some voice commands.

Table 2: Setting value of SK Nugu

Type	Value
User ID	Subonggun
Wifi name	neoidm6
Voice command History	- Hi (안녕) - Please turn on Lee-hi song(이하이 노래 틀어줘) - Who is the starter of Doosan Bears? (두산 베어스 선발투수가 누구야?) - Stop the music (노래 그만) - Tell me the story of a movie parasite(영화 기생충 줄거리 알려줘) - Stop (그만) - How is the weather today? (오늘 날씨 어때?) - Where is the location here? (여기 위치가 어디야?) - How long does it take to chooncheon station? (춘천역까지 얼마나 걸려?)

4.1.2 Following IoT data acquisition procedure

4.1.2.1 Cloud acquisition

① Identify the cloud used as a backend

SKT mentioned, “nugu is not just a speaker, it includes a proprietary AI engine and a cloud server that can handle it” [32]. The presence of a cloud for the SK Nugu service is identified

② Get access to the cloud

The user can access to the cloud data with speakers or mobile apps. If the user does not intentionally log out, the account is still logged in.

③ Ask a potential user (a third party, such as a relative) of the IoT device the credentials

Cloud 3 is excluded because of the basic scenarios for the case study.

④ Request data from a cloud provider

Cloud 4 is excluded because of the basic scenarios for the case study.

⑤ Look for unsecured APIs that does not need authentication to access user data

There are no known unsecured APIs that does not need authentication to access user data.

⑥ Investigate client apps to get accounts and credentials. And use API to access data

a. Look for username and password from client apps file structure.

An analysis of the client app found a username but no password.

b. Look for Credential information such as cookies and tokens from Client apps

The client app for SK nugu exists only for mobile. Data acquisition on the smartphone is performed in Client 2 below. And credential information found from the client app.

4.1.2.2 Client and acquisition

① Browsers and cache

SK nugu don’t have a Browser app

② Smartphone apps forensics

After rooting the smartphone, we used dd on Linux to image the user data. The user data of 27.07 GB (27,078,426,624 GB) was extracted from galaxy note 4 (32 GB). And We then analyzed the image file using MD-RED and found the credential information of the SK NUGU app. As a result of client app analysis, the credential and configuration information found: Authentication_Token, Device_Refresh_Token, Key_SSID_LIST, USER_ID, Wifi_ip_address, Device_Unique_ID, wifi_mac_address, Device_ID, Bluetooth_saved_state, Manufactur_name, Speaker Volume, Moodlight_bright_max, Moodlight_bright_min, Etc.

Figure 10: Credential information from the SK nugu client app(smartphone)

4.1.2.3 Network acquisition

① Web browsers network monitoring tool

SK nugu don't have a browser app

② Man-in-the-Middle attack tools

A. IoT monitoring applications (Smartphone apps or browsers) and Cloud

In this case study, we did a man-in-the-middle attack using Sandroproxy, which intercepts the traffic at the root level of the smartphone app.

a. For intercepting the traffic using Wireshark

We connected the network of the smartphone to the PC, connected the PC and the Internet, and captured packets between the client app and the cloud using Wireshark. (Client app – PC – Cloud)

b. For intercepting the traffic using Man-in-the-Middle tool:

Installing the Sandproxy app on a rooted smartphone and running the SK nugu app can make the Sandroproxy app intercepts the credential information required for client apps to communicate with the cloud and the APIs used to request cloud data. We obtained credential information through this man-in-the-middle attack: Auth-Token, User-Id, Target-Device, T-ID, Phone-Model, etc. And we also found device status.

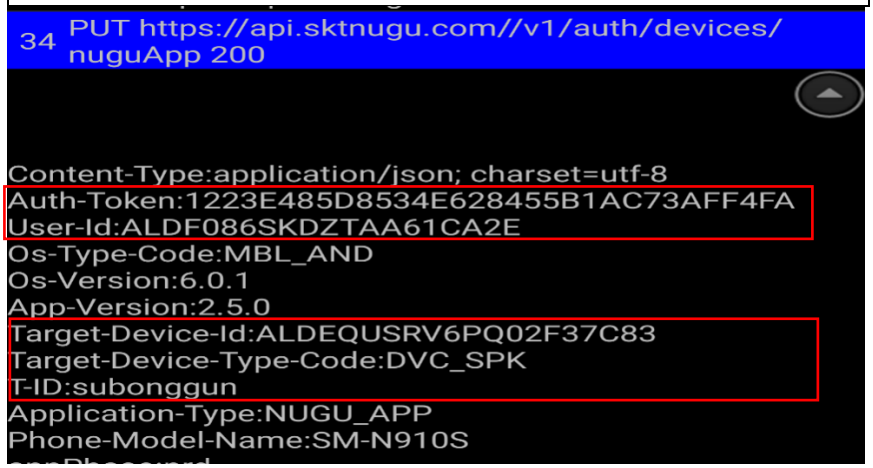

Figure 11: Credential Information from SK nugu cloud using Sandroproxy

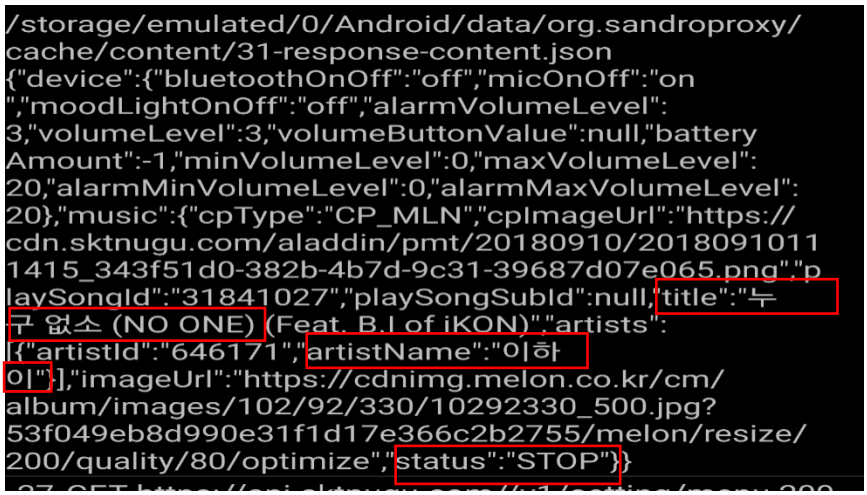

Figure 12: Data from the cloud using Man-in-the-Middle attacks

B. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device:

We connected the network of IoT devices to the PC, connected the PC and the Internet, and captured packets between the IoT device and the cloud using Wireshark. (AI speaker – PC – Cloud)

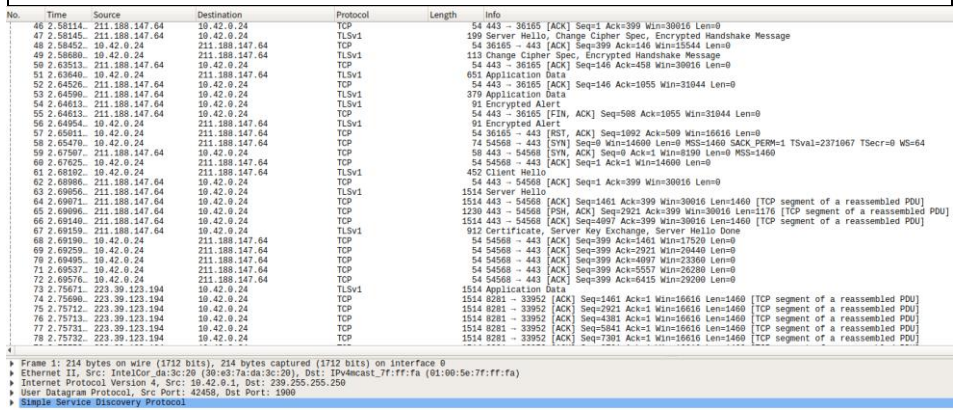

Figure 13: Traffic between SK nugu device and cloud

4.1.2.4 Device acquisition

① Device Level APIs

There is no known device level API.

② Rooting and Imaging

There are no ports available for visual inspection.

③ Destructive methods

Serial port Debug pads were found after removing the rubber pads at the bottom of SK nugu. And On the debug pad, a marking called OTG[6] was identified. After connecting the USB cable to OTG by soldering and connecting it to the PC, The PC device recognized the device named "Google Android Nexus". We used dd to image the user data partition. User data of 6.94 GB (6,936,296,448 bytes) was acquired from the sk nugu device and the time taken was 27 min 81 sec (4,157,135 bytes/sec).

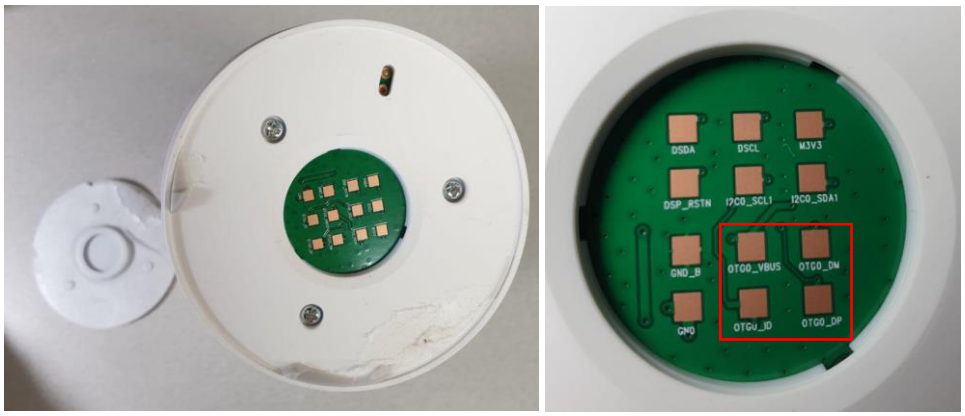

Figure 14: Debug pads found at the bottom of SK nugu

```
latform/bdm/by-name/userdata | busybox nc -l -p 8888
13547454+0 records in
13547454+0 records out
6936296448 bytes transferred in 1668.528 secs (4157135 bytes/sec)
```

Figure 15: User data imaged from SK Nugu device via serial port

[6] OTG - USB On-The-Go, often abbreviated to USB OTG or just OTG, is a specification that a device can perform both master and slave roles.

Chip-off

After disassembling the entire device, the found NAND Flash memory was chip-off, and the entire memory was imaged using the chip reader and MD-NEXT The chip-off was done by Hancom GMD[7]. A total of 8.67 GB (8,672,738,304 bytes) was extracted from the chip-off and took 11min 26 sec (12,595,212 bytes / sec).

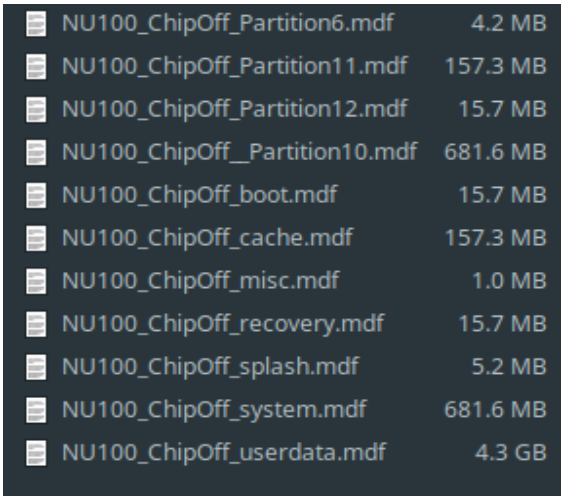

Figure 16: SK nugu Data extracted from chip-off

4.1.3 Case study result of SK Nugu

The results of SK Nugu's data acquisition following the IoT data acquisition procedure are shown in Table 3 below. The cloud side was able to get the device state and some command history with the credentials obtained from the client app and the API obtained from the network acquisition. On the network side, we can get the APIs and credential information by man-in-the-middle attack using Sandroproxy, and we can capture network traffic between the client app and cloud, network traffic between device and cloud using Wireshark. Network traffic was encrypted using protocol TLSv1, and network traffic continued to occur even when there was no command on the IoT device.

[7] **Hancom GMD** is located in Korea, researching mobile and digital forensics and developing forensic solutions.

The client side was able to acquire 27.07 GB of user data on the smartphone. and we found credential information from the client app in the user data. The acquired credential information from client app included a refresh token, which can be used to create a new authentication token. For example, if an investigator has acquired an expired authentication token, a new authentication token can be created with a refresh token and used for credentials. The device side was able to acquire 6.94GB of user data from the serial port found at the bottom of the device and 8.67GB from the memory chip. In addition, data obtained from the device side included the credential information for communicating with the cloud.

Measuring the amount of data was only possible with data from the device, and the amount of data on the cloud or the network side could not represent a specific value because it depends on the API or depends on the network traffic capture time. If SK Nugu's APIs can be obtained in advance by conducting previous research on SK Nugu, the investigator can acquire cloud data using the APIs immediately after obtaining the credential information without Network acquisition. In addition, because the credential information was also found on the device, if the data cannot be acquired from the client side, the credential information can be acquired from the device.

Table 3: Result of SK Nugu

Acquisition	Source	Size of Data
Cloud	APIs	Device status, some command history
Network	Man-in-the-Middle	APIs, Credential information.
	Packet analysis tool	Pcap
Client	Smart phone	27.07GB
		Device configuration, Credential information
Device	Serial port	6.94GB
	Chip-off	8.67GB

4.2 NAVER CLOVA

Naver Clover is an artificial intelligence awareness service that uses deep learning techniques similar to most other artificial secretarial services. Naver is the most used search engine service in Korea[33]. Clover provides various services such as search, weather, music recommendation and playback, translation, IoT control and free talking using Naver searching engine. In order to use Naver Clova, initial settings are required, and initial settings can only be made in conjunction with a smartphone. And since these services are provided in partnership with other vendors, sign in and login are required for each service in order to use other services. Once logged in, if the user does not intentionally log out, the application will continue to log in. Naver Clova's account is linked with Naver account, and if there is a Naver app on the user's smartphone, it will be automatically linked.

4.2.1 Naver Clova Setting

In order to perform a case study, the actual IoT device must have data. We initialized it with the following account and generated the data with some voice commands.

Table 4: The setting value of Naver Clova

Type	Value
User ID	dog1860
Wifi name	neoidm6
Voice command History	- How is the weather toady? (오늘 날씨 어때?) - Let me know the weather today(오늘 날씨 알려줘) - Hi(안녕) - turn on a song(노래 하나 들려줘) - What's after the Masters? (석사 다음에 뭐야?) - Find a good restaurant in Chuncheon (춘천 맛집 찾아줘) - Smoking is bad? (흡연이 몸에 나빠?) - Tell me the story of a movie parasite (영화 기생충 줄거리 알려줘) - Where is the location here?(여기 위치가 어디야?) - How long does it take to get to Chuncheon Station?(춘천역까지 얼마나 걸려?)

4.2.2 Following IoT data acquisition procedure

4.2.2.1 Cloud acquisition

① Identify the cloud used as a backend

Naver Clover is a cloud-based AI service platform. And it can connect to other IoT products using the cloud.

② Get access to the cloud

The user can access to the cloud with speakers or mobile apps. If the user does not intentionally log out, the account is still logged in.

③ Ask a potential user (a third party, such as a relative) of the IoT device the credentials

Cloud 3 is excluded because of the basic scenarios for the case study.

④ Request data from a cloud provider

Cloud 4 is excluded because of the basic scenarios for the case study.

⑤ Look for unsecured APIs that does not need authentication to access user data

Official APIs are provided to users to develop with Naver-services, but in order to use them, the user must subscribe to the service on the developer page and receive a token for the official API

⑥ Investigate client apps to get accounts and credentials. And use API to access data

The client app for Naver Clova exists only for mobile. Data acquisition on the smartphone is performed in Client 2 below.

a. Look for username and password from client apps file structure.

An analysis of the client app found a username but no password.

b. Look for Credential information such as cookies and tokens from Client apps

The client app for Naver Clova exists only for mobile. Data acquisition on the smartphone is performed in Client 2 below. And credential information found from the client app.

4.2.2.2 Client acquisition

① Browsers and cache

Naver Cloud don't have a Browser app

② Smartphone apps forensics

After rooting the smartphone, we used dd on Linux to image the user data. The user data of 27.07 GB (27,078,426,624 GB) was extracted from galaxy note 4 (32 GB). And We then analyzed the image file using MD-RED and found the credential information of the Naver Clova app. The credential information found: RefreshToken, AccessToken

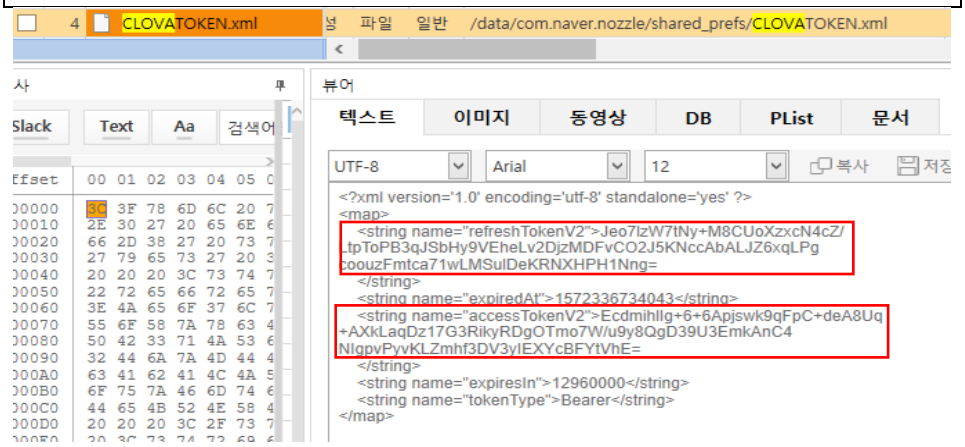

Figure 17: Credential information(token) from the Naver Clova client app

4.2.2.3 Network acquisition

① Web browsers network monitoring tool

The Naver Clova don't have a browser app

② Man-in-the-Middle attack tools

A. IoT monitoring applications (Smartphone apps or browsers) and Cloud

a. For intercepting the traffic using Wireshark

We connected the network of smartphone to the PC, connected

the PC and the Internet, and captured packets between the client app and the cloud using Wireshark. (Client app – PC – Cloud)

b. For intercepting the traffic using Man-in-the-Middle tool

Installing the Sandproxy app on a rooted smartphone and running the Naver Clova app can make the Sandroproxy app intercepts the credential information required for client apps to communicate with the cloud and the APIs used to request cloud data. We obtained credential information through this man-in-the-middle attack: Bearer token, Device-Id, Client Model, APIs etc. And several APIs have been discovered that return device activity information values and the user command history (Figure 18).

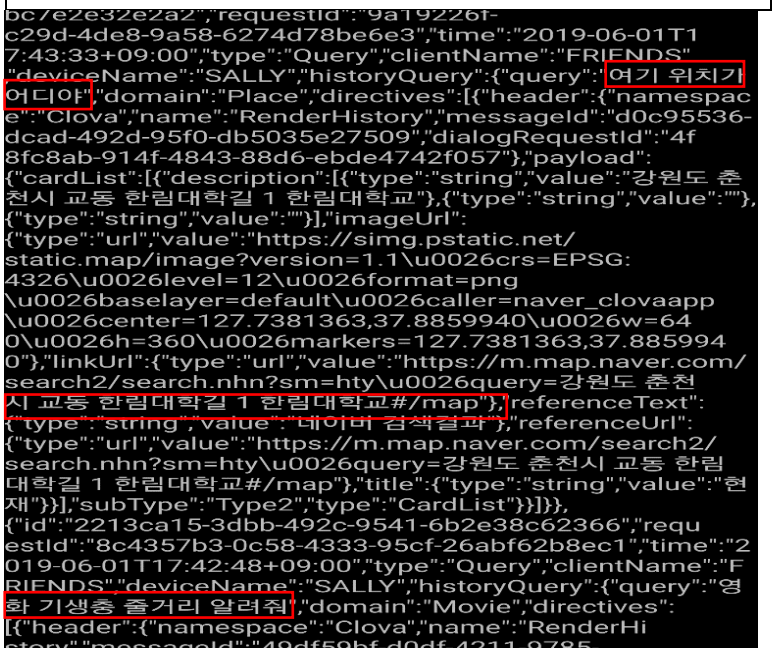

Figure 18: Data of Naver Clova cloud using Man-in-the-Middle attacks

B. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device

We connected the network of IoT devices to the PC, connected the PC and the Internet, and captured packets between the IoT device and the cloud using Wireshark. (AI speaker – PC – Cloud)

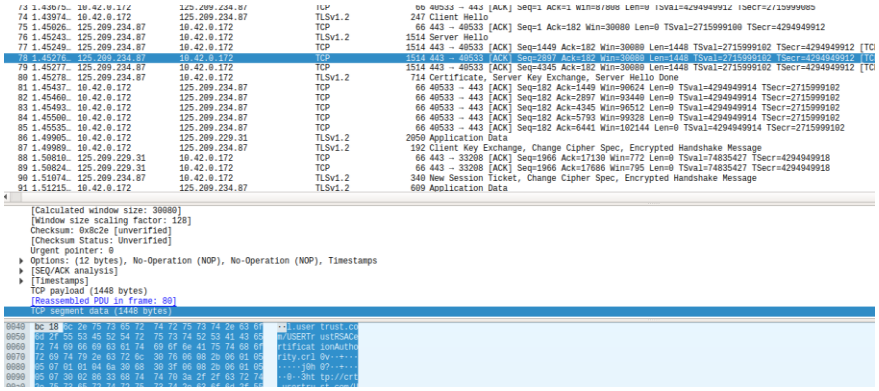

Figure 19: Traffic between Naver Clova device and cloud

4.2.2.4 Device acquisition

① Using Device Level APIs

There is no known device-level APIs.

② Rooting and Imaging

There are no ports available for visual inspection.

③ Destructive methods

Chip-off After disassembling the entire device, the found NAND Flash memory was chip-off, and the entire memory was imaged using the chip reader and MD-NEXT The chip-off was done by Hancom GMD. A total of 4.3 GB (4,294,967,295 bytes) was extracted from the chip-off and took 4min 4 sec (17,602,324 bytes / sec).

Figure 20: Naver Clova Data extracted from chip-off

4.2.3 Case study result of Naver Clova

The results of Naver Clova's data acquisition following the IoT data acquisition

procedure are shown in Table 5 below. The cloud side was able to obtain user command history data with the API obtained from the network acquisition and the credential information obtained from the client app. The acquired credential information included a refresh token, which can be used to create a new authentication token. For example, if an investigator has acquired an expired authentication token, a new authentication token can be created with a refresh token and used for credentials. On the network side, we can get the APIs and credential information by man-in-the-middle attack using Sandroproxy, and we can capture network traffic between the client app and cloud, network traffic between device and cloud using Wireshark. Network traffic was encrypted using protocol TLSv1.2, and network traffic continued to occur even when there was no command on the IoT device. The client side was able to acquire 27.07 GB of user data on the smartphone. and we found credential information from the client app in the user data. The acquired credential information from client app included a refresh token, which can be used to create a new authentication token. For example, if an investigator has acquired an expired authentication token, a new authentication token can be created with a refresh token and used for credentials. And The device side was able to acquire 4.3GB from the memory chip. In addition, data obtained from the device side included the credential information for communicating with the cloud.

Measuring the amount of data was only possible with data from the device, and the amount of data on the cloud or the network side could not represent a specific value because it depends on the API or depends on the network traffic capture time. If Naver Clova's APIs can be obtained in advance by conducting previous research on Naver Clova, the investigator can acquire cloud data using the APIs immediately after obtaining the credential information without Network acquisition. In addition, because the

credential information was also found on the device, if the data cannot be acquired from the client side, the credential information can be acquired from the device.

Table 5: Result of Naver Clova

Acquisition	Source	Data
Cloud	API	Command History
Network	Man-in-the-Middle	APIs Credential information.
	Packet analysis tool	Pcap
Client	Smart phone	27.07GB
		Credential information
Device	Chip-off	4.3GB

4.3 KAKAO MINI

The Kakao Mini is an artificial intelligence (AI) speaker with an artificial intelligence platform 'Kakao I'. Kakao offers various services such as searching service, calling a taxi, Smart Home IoT control, and music service, as well as KakaoTalk, the number one mobile messenger in Korea. The Kakao mini is linked to controlling various services provided by Kakao by voice. In order to use Kakao mini, initial settings are required, and initial settings can only be made in conjunction with a smartphone. And since these services are provided in partnership with other vendors, sign in and login are required for each service in order to use other services. Once logged in, if the user does not intentionally log out, the application will continue to log in. Kakao mini's account is linked with Kakao account, and if there is a kakao app such as kakao talk, kakao taxi on the user's smartphone, it will be automatically linked.

According to Kakao's data policy, Kakao data is not stored in the cloud. And the message, the taxi call, the memo, the schedule, etc. are sent and stored in the service app such as the KakaoTalk app or the Kakao taxi app. For example, if user command to make memo or schedule with a voice command to kakao mini, the note and schedule are left in the user's chat in KakaoTalk. So, when users check data, they have to go into the app to check.

4.3.1 Kakao Mini Setting

In order to perform a case study, the actual IoT device must have data. We initialized it with the following account and generated the data with some voice commands

Table 6: The setting value of Kakao mini

Type	Value
User ID	jangsubong@gmail.com

Wifi name	neoidm6
Voice command History	- How is the weather toady? (오늘 날씨 어때?) - Let me know the weather today (오늘 날씨 알려줘) - Hi(안녕) - turn on a song(노래 하나 들려줘) - What's after the Masters? (석사 다음에 뭐야?) - Find a good restaurant in Chuncheon (춘천 맛집 찾아줘) - Smoking is bad? (흡연이 몸에 나빠?) - Tell me the story of a movie parasite (영화 기생충 줄거리 알려줘) - Where is the location here? (여기 위치가 어디야?) - How long does it take to get to Chuncheon Station?(춘천역까지 얼마나 걸려?)

4.3.2 Following IoT data acquisition procedure

4.3.2.1 Cloud acquisition

① Identify the cloud used as a backend

Kakao mini is a cloud-based AI service platform. And it can connect to other service and IoT products using the cloud. However, Kakao Mini has a data policy that does not store data in the cloud.

② Get access to the cloud

The user can access to the cloud with speakers or mobile apps. If the user does not intentionally log out, the account is still logged in.

③ Ask a potential user (a third party, such as a relative) of the IoT device the credentials

Cloud 3 is excluded because of the basic scenarios for the case study.

④ Request data from a cloud provider

Cloud 4 is excluded because of the basic scenarios for the case study.

⑤ Look for unsecured APIs that does not need authentication to access user data

Official APIs are provided to users to develop with kakao-services , but in order to use them, the user must subscribe to the service on the developer page and

receive a token for the official API

⑥ Investigate client apps to get accounts and credentials. And use API to access data

The client app for kakao mini exists only for mobile. Data acquisition on the smartphone is performed in Client 2 below. And there is also a browser app for the Kakao service that allows the user to view the login history, connected device history, and linked service history for kakao account. But this browser app has nothing to do with kakao mini device operation.

a. Look for username and password from client apps file structure.

An analysis of the client app found a username but no password.

b. Look for Credential information such as cookies and tokens from Client apps

The credential and configuration information was found from the client app

4.3.2.2 Client acquisition

① Browsers and cache

Kakaomini don’t have a Browser app

② Smartphone apps forensics

After rooting the smartphone, we used dd on Linux to image the user data. The user data of 27.07 GB (27,078,426,624 GB) was extracted from galaxy note 4 (32 GB). And We then analyzed the image file using MD-RED and found the credential information and personal information of the kakao mini app (see figure 21). Credential information: Cookies, Refresh Token, Access Token Personal information: name, phone number, birthday, etc.

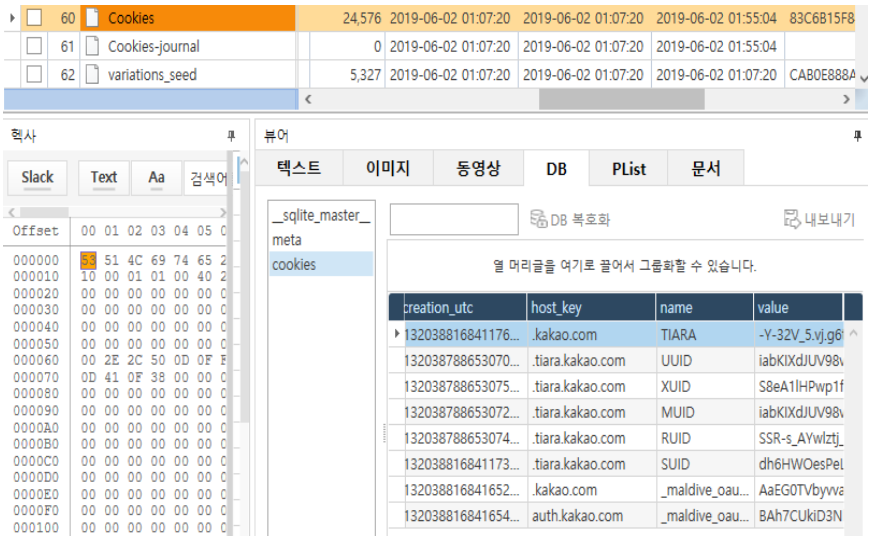

Figure 21: Credential Information of kakao mini from client app(mobile)

4.3.2.3 Network acquisition

① Web browsers network monitoring tool

There is no web browser app to access the Kakao mini data, but there is a web page (kakao account manage page) that shows the record of the Kakao account login, service connection, device connection. We used chrome's devtool to get the API related to the data requested from the web browser and confirmed that the cookie was used to request the data. We were able to find the cookie used in the user data of the Chrome browser on the PC using a browser. When we requested data using the API and cookie, we confirmed that the data returned as a JSON value. This browser, which manages a kakao account, keeps data only for up to 90 days.

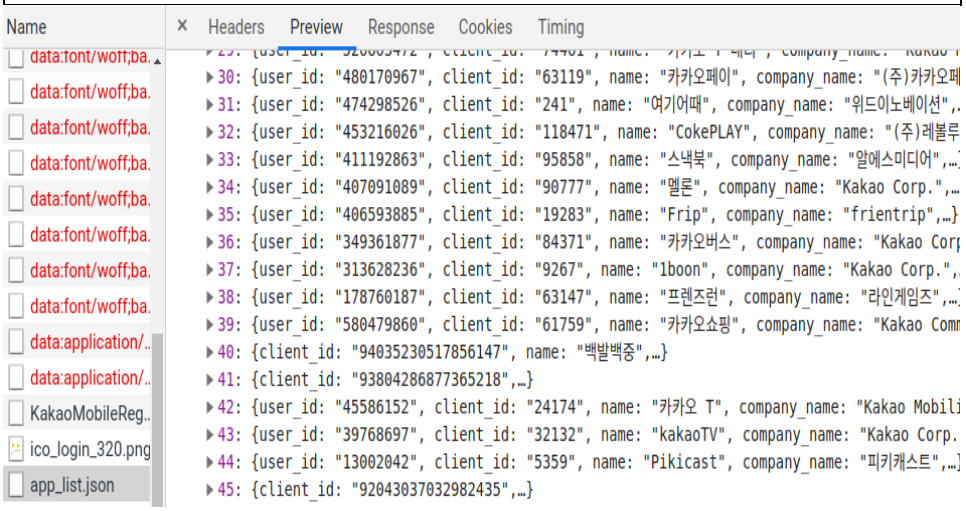

Figure 22: Data that can be verified using devtool

② Man-in-the-Middle attack tools

A. IoT monitoring applications (Smartphone apps or browsers) and Cloud

In this case study, we did a man-in-the-middle attack using Sandroproxy, which intercepts the traffic at the root level of the smartphone app.

a. For intercepting the traffic using Wireshark

We connected the network of smartphone to the PC, connected the PC and the Internet, and captured packets between the client app and the cloud using Wireshark. (Client app – PC – Cloud)

b. For intercepting the traffic using Man-in-the-Middle tool:

The Kakao mini client app (mobile) is designed to prevent man-in-the-middle attacks, so the apps have stopped working when the man-in-the-middle attack tool Burb suit and Sandroproxy are used.

B. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device:

We connected the network of IoT devices to the PC, connected the PC and the Internet, and captured packets between the IoT device and the cloud using Wireshark. (AI speaker – PC – Cloud)

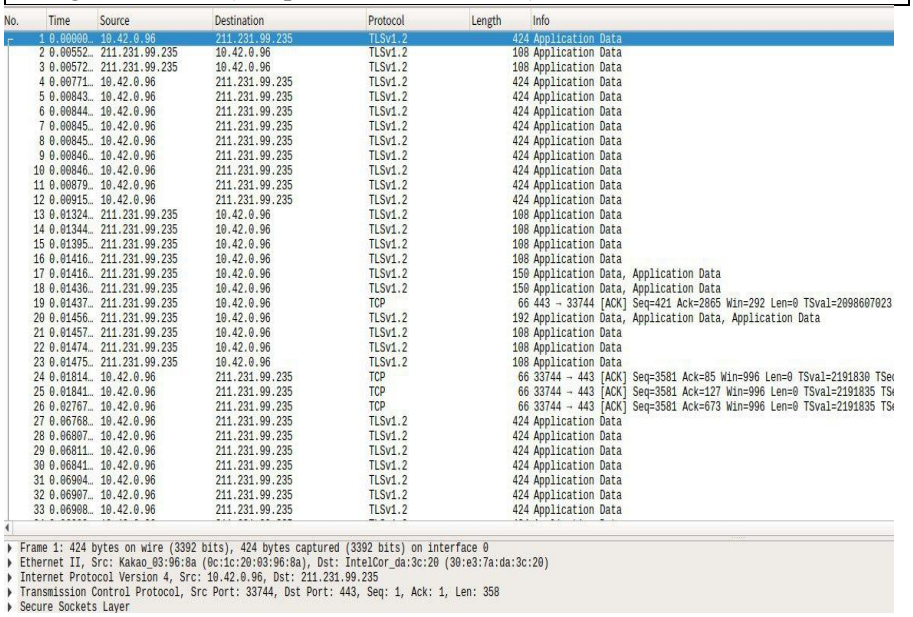

Figure 23: Traffic between kakao mini device and cloud

4.3.2.4 Device acquisition

① Using Device Level APIs

There is no known device-level APIs

② Rooting and Imaging

There is a USB terminal on the outside of Kakao mini. According to the manual, the USB terminal is only for charging purposes. We connected to a PC or connected a USB memory containing music, but neither PC nor Kakao Mini responded.

③ Destructive methods

Serial port Debug pads can be seen by removing the rubber at the bottom of the instrument. However, there is no indication of the role of each port on the PCB. The connection is not possible because the role of each port is unknown. On the other side of the PCB, there are two ports labelled USB_BOOT. We guessed the two ports as DATA signal lines and connected them to the computer, but it did not work. It seems like Amazon Alexa blocked logically external access.

Figure 24: Attempt to connect debug port and serial port of kakao mini

Chip-off

After disassembling the entire device, the found NAND Flash memory was chip-off, and the entire memory was imaged using the chip reader and MD-NEXT. The chip-off was done by Hancom GMD. A total of 3.9 GB (3,904,897,024 bytes) was extracted from the chip-off and took 4min 34 sec (14,251,448 bytes / sec).

Figure 25: Kakao mini Data extracted from chip-off

4.3.3 Case study result of Kakao mini

The results of Kakao Mini's data acquisition following the IoT data acquisition procedure are shown in Table 7 below. According to the data policy of Kakao, the cloud of Kakao does not store data, so data acquisition by cloud acquisition is impossible. And through the case study, we did not get data from the cloud. However, we were able to obtain login history, connected services, and connected devices from a web page that manages the kakao account. The details are described below on the client side. On the network side, we can capture network traffic between the client app and cloud, network traffic between device and cloud using Wireshark. Network traffic was encrypted using protocol TLSv1.2, and network traffic continued to occur even when there was no command on the IoT device. However, man-in-the-middle attacks were not possible because of the security enhancements of smartphone client apps. Client apps installed on smartphone detect the behavior of the burb suit or Sandroproxy and stop the client

app. The client side was able to acquire 27.07 GB of user data on the smartphone. And while Kakao Mini does not provide a browser app, Kakao provides a web page for account management, and we can access that page through a web browser. The Kakao Account Management page provides the login history, connected services, and connected device history for a Kakao account. We found the API used to request data from this page using the Chrome browser's devtool and found the cookie that was generated when logging in from the PC. This page has nothing to do with controlling or setting up the Kakao mini, but we can see which apps are available for the Kakao mini and which types of mobile phones have been linked to the Kakao mini. On this page, when a service app that can be used on a Kakao mini is identified, the app must also be investigated. For example, Kakao mini will load a Kakao taxi app installed on a smartphone when using a Kakao taxi. Therefore, when the connected service is confirmed, we have to analyze the service app. On the device side, the USB on the exterior of the Kakao mini showed no response when connected to a PC or USB memory for charging, and we tried connecting to the port marked USB BOOT on the PCB board inside the device, but we could not get any results. and we were able to acquire 3.9GB from the memory chip off.

Measuring the amount of data was only possible with data from the device, and the amount of data on the cloud or the network side could not represent a specific value because it depends on the API or depends on the network traffic capture time. We could not get the API for Kakao Mini. But if Kakao mini's APIs can be obtained in advance by conducting previous research on Kakao Mini, the investigator may acquire cloud data using the APIs using the credential information from the client app. But Data may not exist in the cloud as the Kakao data policy does. However, we acquired several data. first,

the data from the Kacao Mini Device contained an audio file in which the Kakao mini responded to the user's command. That is, the device contains data that can not be obtained from the cloud, and this data can be potentially important evidence. The second, kakao account management page allows the user to view the interlocked services and account used history, so the investigator can investigate the apps that connected with kakao account

Table 7: Result of Kakao mini

Acquisition	Source	Data
Cloud	API for account management	History of login, Connected service, Connected device
Network	Web networking monitoring tool	API for account management
	Packet analysis tool	Pcap
Client	Smart phone	27.07GB
		Credential Information
	Browser in PC	Cookie
Device	Chip-off	3.9GB

4.4 GIGA GENIE

Giga genie is an AI speaker product with set-top box feature launched in January 2017 at KT(Korea telecom which is one of the major telecommunications company in Korea). Unlike other AI speakers, most of the settings are available via a remote controller. And it can be initialized as a smartphone client app. There are a media-only USB port and Micro SD slot, HDMI, S / PDIF and LAN on the outside of the device. Giga genie provides services such as search, finance, weather, memo, schedule, music, IoT device control, etc., and TV can be controlled by voice. And since these services are provided in partnership with other vendors, sign in and login are required for each service in order to use other services. Once logged in, if the user does not intentionally log out, the application will continue to log in. Giga Genie's account can be used in conjunction with a KT account and can be used in conjunction with Kakao, Naver, Facebook, and mobile phone authentication.

4.4.1 Giga Genie Setting

In order to perform a case study, the actual IoT device must have data. We initialized it with the following account and generated the data with some voice commands

Table 8: The setting value of Giga Genie

Type	Value
User ID	subonggun
Wifi name	neoidm6
Voice command History	-Hi (안녕) -Play Lee Hi song (이하이 노래 틀어줘) -Who is the starter of Doosan Bears? (두산 베어스 선발 투수가 누구야?) - stop musing (노래 그만) - Tell me the story of a movie parasite (영화 기생충 줄거리 알려줘) - stop (그만) - how is the weather today? (오늘 날씨 어때?) - Where is the location here? (여기 위치가 어디야?) - How long does it take to get to Chuncheon Station? (춘천역까지 얼마나 걸려?)

4.4.2 Following IoT data acquisition procedure

4.4.2.1 Cloud acquisition

① Identify the cloud used as a backend

Giga genie is a cloud-based AI speaker.

② Get access to the cloud

The user can access to the cloud with speakers or mobile apps. If the user does not intentionally log out, the account is still logged in.

③ Ask a potential user (a third party, such as a relative) of the IoT device the credentials

Cloud 3 is excluded because of the basic scenarios for the case study.

④ Request data from a cloud provider

Cloud 4 is excluded because of the basic scenarios for the case study.

⑤ Look for unsecured APIs that does not need authentication to access user data

Official APIs are provided to users to develop with Giga genies services, but in order to use them, the user must subscribe to the service on the developer page and receive a token for the official API.

⑥ Investigate client apps to get accounts and credentials. And use API to access data

Credential information was obtained from the client app, but it was not possible to gain access to the cloud data because the API was not available.

a. Look for username and password from client apps file structure.

An analysis of the client app found a username but no password.

b. Look for Credential information such as cookies and tokens from Client apps.

The credential information found: Refresh Token, Access Token.

4.4.2.2 Client acquisition

① Browsers and cache

Giga genie doesn't have a Browser app

② Smartphone apps forensics

After rooting the smartphone, we used dd on Linux to image the user data. The user data of 27.07 GB (27,078,426,624 GB) was extracted from galaxy note 4 (32 GB). And We then analyzed the image file using MD-RED and found the credential information (Refresh Token, Access Token) of the Giga genie app (figure 26).

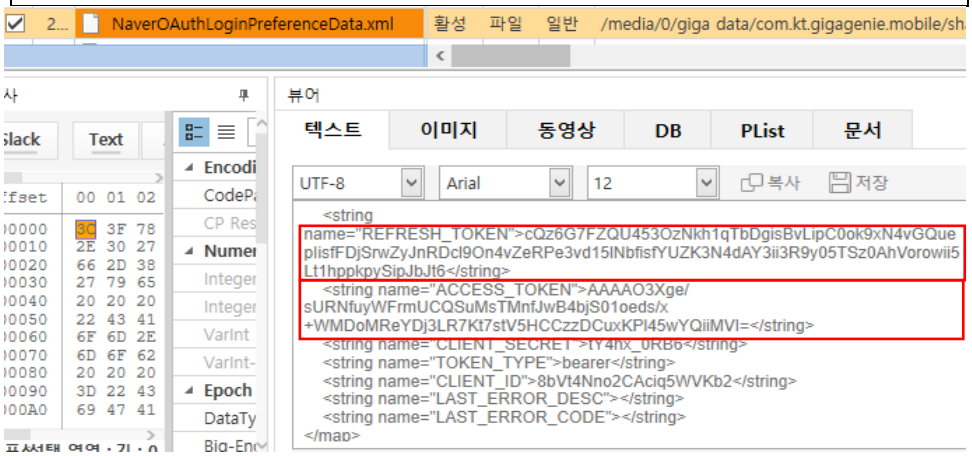

Figure 26: Credential Information of Giga Genie from the client app(smartphone)

4.4.2.3 Network acquisition

① Web browsers network monitoring tool

Giga genie cloud doesn't have a browser app

② Man-in-the-Middle attack tools

A. IoT monitoring applications (Smartphone apps or browsers) and Cloud

a. For intercepting the traffic using Wireshark

The Giga genie client app(mobile) - cloud traffic were collected by Wireshark.

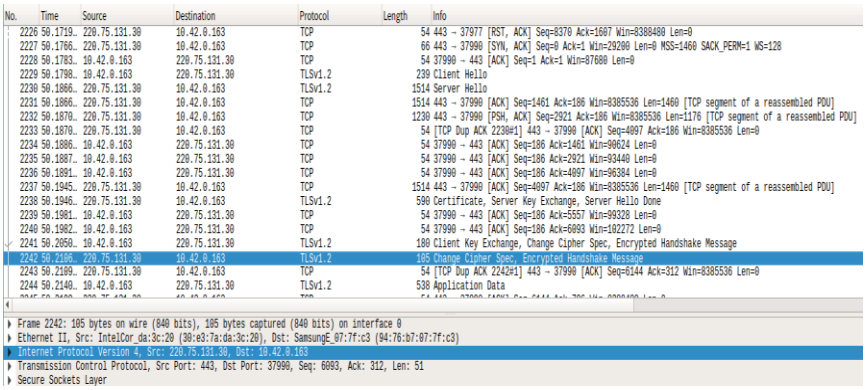

Figure 27: Network traffic between the client app and cloud

b. For intercepting the traffic using Man-in-the-Middle tool

The Giga genie client app (mobile) is designed to prevent man-in-the-middle attacks, so the apps have stopped working when the man-in-the-middle attack tool Burb suit and Sandroproxy are used.

B. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device

The Giga genie devices-cloud traffic was collected by Wireshark.

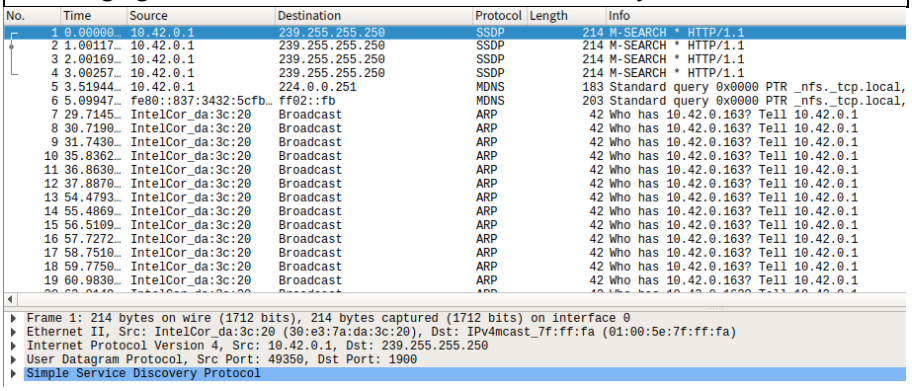

Figure 28: Network traffic between Giga genie device and cloud

4.4.2.4 Device acquisition

① Using Device Level APIs

There is no known device-level API

② Rooting and Imaging

According to the instructions, the USB terminal and Micro SD slot are for media playback only. We connected the USB terminal to the PC, but the PC did not

read anything.

③ Destructive methods

Serial Port

After disassembling the Giga genie, the suspect part of the serial port is found, but serial extraction is not done because there is no mark on the port.

Chip-off

After disassembling the entire device, the found NAND Flash memory was chip-off, and the entire memory was imaged using the chip reader and MD-NEXT. The chip-off was done by Hancom GMD. A total of 4.3 GB (4,294,967,567 bytes) was extracted from the chip-off and took 5 min 41 sec (12,595,212 bytes / sec).

Figure 29: Giga Genie Data extracted from chip-off

4.4.3 Case study result of Giga genie

The results of Giga genie's data acquisition following the IoT data acquisition procedure are shown in Table 9 below. The cloud side was unable to acquire any data. The client app collected credential information, but the API was not available. On the network side, we can capture network traffic between the client app and cloud, network traffic between device and cloud using Wireshark. Network traffic was encrypted using protocol TLSv1.2, and network traffic continued to occur even when there was no command on the IoT device. However, man-in-the-middle attacks were not possible because of the security enhancements of smartphone client apps. Client apps installed on smartphone detect the behavior of the burb suit or Sandroproxy and stop the client app. The client side was able to acquire 27.07 GB of user data on the smartphone. And The device side was able to acquire 4.3GB from the memory chip. In addition, data

obtained from the device side included the credential information for communicating with the cloud.

Measuring the amount of data was only possible with data from the device, and the amount of data on the cloud or the network side could not represent a specific value because it depends on the network traffic capture time. And we couldn't get API from Network acquisition. If Giga genie's APIs can be obtained in advance by conducting previous research on Giga genie, the investigator can acquire cloud data using the APIs immediately after obtaining the credential information without Network acquisition. In addition, because the credential information was also found on the device, if the data cannot be acquired from the client side, the credential information can be acquired from the device. After analyzing the data obtained from the device, several user voice recording files were found in the data obtained from the device, and several command histories were found in the client app cache. It is not a whole history, but there is a potential for evidence.

Table 9: Result of Giga genie

Acquisition	**Source**	**Data**
Cloud	-	-
Network	Packet analysis tool	Pcap
Client	Smart phone	27.07GB
		Credential information
Device	Chip-off	4.3GB

4.5 GOOGLE HOME MINI

Google Home is deep learning based artificial intelligence speaker from Google. It is the hardware version of Google assistant which is located in Android phone. It provides various services based on Google services and is the first overseas product of artificial intelligent speakers to provide services to Korea. In order to use Google home min, initial settings are required, and initial settings can only be made in conjunction with a smartphone. And since these services are provided in partnership with other vendors, sign in and login are required for each service in order to use other services. Once logged in, if the user does not intentionally log out, the application will continue to log in. The Google Home Mini is based on the language user set. For example, After the user sets Korean and English, when the user commands in Korean, Google responds in Korean. When the user commands in English, Google responds in English.

4.5.1 Google home mini Setting

In order to perform a case study, the actual IoT device must have data. We initialized it with the following account and generated the data with some voice commands

Table 10: The setting value of Google home mini

Type	Value
User ID	simonhallym@gmail.com
Wifi name	neoidm6
Voice command History	* Since Google Home Mini supports a variety of languages, it has mixed English and Korean commands. In the history below, what is in front is the commanded language, and the parentheses are translations into English.
	-볼륨 맥시멈(volume maximum) - Hi - What is your name? - How can I get to the chooncheon station from here? -춘천역까지 얼마나 걸려? (How long does it take to get to Chuncheon Station) - Hi, How are you? - Where is Harvard University?

4.5.2 Following IoT data acquisition procedure

4.5.2.1 Cloud acquisition

① Identify the cloud used as a backend

Google home mini, also known as artificial intelligence speakers, is a cloud-based Google Assistance service.

② Get access to the cloud

The user can access cloud data using speaker and client apps (smartphone app and browser app). For smartphone apps, if the user does not intentionally log out, the account remains logged in. And in the case of the browser app, it may be different for each user's configuration in the browser.

③ Ask a potential user (a third party, such as a relative) of the IoT device the credentials

Cloud 3 is excluded because of the basic scenarios for the case study.

④ Request data from a cloud provider

Cloud 4 is excluded because of the basic scenarios for the case study.

⑤ Look for unsecured APIs that does not need authentication to access user data

Google provides official APIs. But all the API need authentication. So there is no known unsecured APIs.

⑥ Investigate client apps to get accounts and credentials. And use API to access data

We identified the API from the Browser app and got the cookie from the Chrome browser on my PC. We acquired user history data using APIs and cookies.

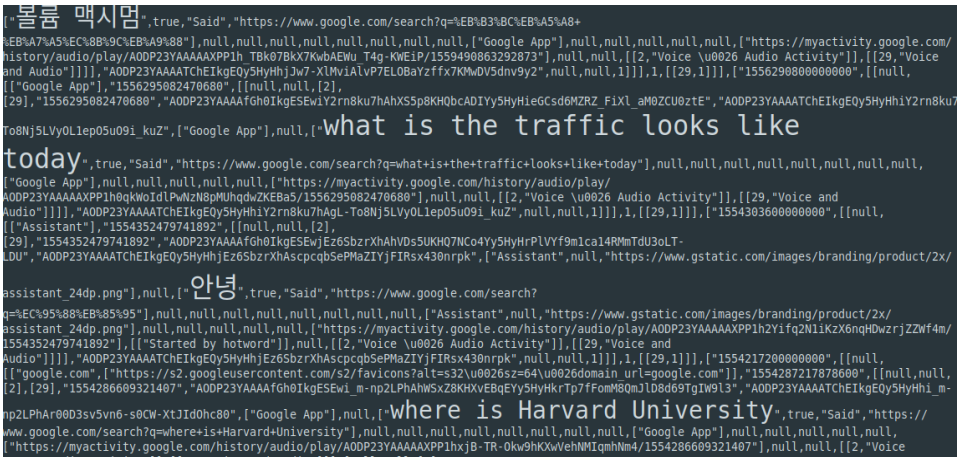

Figure 30: Google home mini history return value obtained through API

a. Look for username and password from client apps file structure.

An analysis of the client app found a username but no password.

b. Look for Credential such as cookies and tokens from Client apps

The cookie was obtained from the client app. The process of obtaining cookies is described below in client acquisition.

4.5.2.2 Client acquisition

① Browsers and cache

Since google home mini has access to the command history through a browser app, it needs to extract the cookies that remain in the PC browser app to use the API. In the path below we got the cookies stored in the Chrome browser. And we checked the cookie file with DB VIEW, and it contains the cookies that were used to login to google.com. Path - *C:\Users\jangsubong\AppData\Local\Google\Chrome\User Data\Default\Cookies*

② Smartphone apps forensics

After rooting the smartphone, we used dd on Linux to image the user data. The user data of 27.07 GB (27,078,426,624 GB) was extracted from galaxy note 4 (32 GB). And We then analyzed the image file using MD-RED and found the cookie in Google home mini app

4.5.2.3 Network acquisition

① Web browsers network monitoring tool

If we access *myactivity.google.com* using the browser, we can see user's activity history related to Google using user's account and it contains google home mini history. If there was a voice command through the client google app, the voice is recorded. Using chrome devtool, we could find an API that returns google mini history value.

② Man-in-the-Middle attack tools

A. IoT monitoring applications (Smartphone apps or browsers) and Cloud

a. For intercepting the traffic using Wireshark

The Google client app(browser) - cloud traffic were collected by Wireshark

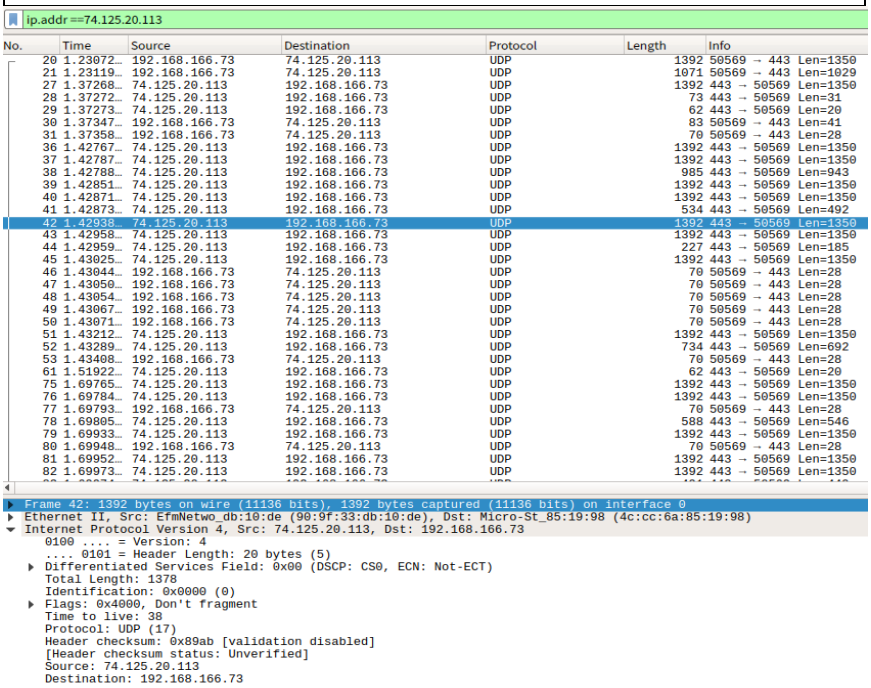

Figure 31: Google browser app-cloud traffic

b. For intercepting the traffic using Man-in-the-Middle tool:

Installing the Sandproxy app on a rooted smartphone and running the Google home mini app can make the Sandroproxy app intercepts the credential information required for client apps to communicate with the cloud and the APIs used to request cloud data. We obtained credential information through this man-in-the-middle attack: cookies, several APIs have been discovered that return device activity information values and the user command history(Figure 32).

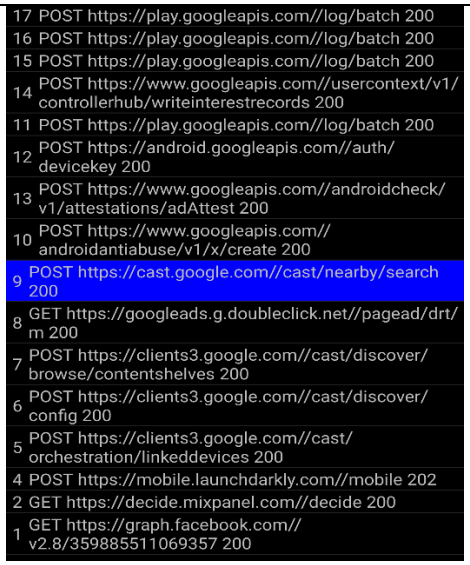

Figure 32: APIs from google cloud for google home mini

B. IoT Device/Hub and Cloud - intercepting the traffic between the IoT device

The Google home mini device - cloud traffic were collected by Wireshark

4.5.2.4 Device acquisition

① Using Device Level APIs

There is no known device-level APIs

② Rooting and Imaging

There are no ports available for visual inspection.

③ Destructive methods

Chip-off To acquire data from a chip, we must use a chip reader device. The chip reader device depends on the size of the chip, but we could not get it because we did not have a reader device for google home mini memory chip size.

4.5.3 Result of Google Home mini

The results of Google home mini's data acquisition following the IoT data acquisition procedure are shown in Table 11 below. The cloud side was able to obtain user history data from the Google home mini using the cookies from the client acquisition and the API from the network acquisition. On the network side, we can capture network traffic between the client app and cloud, network traffic between device and cloud using Wireshark. Network traffic was using protocol UDP, and network traffic continued to occur even when there was no command on the IoT device. We have also acquired an API that responds to user history in browser apps using the web browser network monitoring tool. And in the Man-in-the-Middle attack, APIs related to authenticating, weather, and user history were found in the client app on the smartphone. The client side was able to acquire 27.07 GB of user data on the smartphone. On the device side, it was only possible to chip off, but it was impossible to extract data because we don't have a chip reader for google home mini. The chip size of google home mini differs from the chip size of other AI speakers, so we could not use our chip reader.

Measuring the amount of data was only possible with data from the device, and the amount of data on the cloud or the network side could not represent a specific value because it depends on the network traffic capture time and the amount of data that the cloud has. Data from Google home mini is linked to google account and can be viewed on the Google activity web page. The data is the user's command history and voice

recording files. Investigator can also see which services are interlocked, which can help the investigator identify other IoT devices. The credential information for acquiring this data is found on both browser and smartphone app. If google home mini's APIs can be obtained in advance by conducting previous research on google home mini, the investigator can acquire cloud data using the APIs immediately after obtaining the credential information from client acquisition without network acquisition and device acquisition.

Table 11: Result of Google Home mini

Acquisition	Source	Data
Cloud	API	User Activity history Recorded user voice
Network	Web monitoring tool	API
	Packet analysis tool	Pcap
Client	Smart phone	27.07GB
		Credential information
	Browser	Credential information
Device	-	-

4.6 Result of case studies

The purpose of this case study is following the IoT acquisition procedure of cloud, client, network, device presented in chapter 3 to measure how much data can be obtained from the IoT device and whether the data is acquired. As a result of the case study, it was possible to acquire data from all devices as shown in Table 12. SK Nugu, Naver Clova, and Kakao Mini were able to obtain data from all acquisition procedures. SK Nugu, Naver Clova, were able to obtain data from all acquisition procedures. And it was impossible to acquire data from Giga Genie's cloud acquisition, Kakao Mini's cloud acquisition and google home mini's device acquisition. In the case of Kakao mini, when a man-in-the-middle attack comes, the client app detects and stops the app from running. So, we could not get the API to access cloud data. However, Kakao provides an account management web page such as user account login history, connected device history, and connected services. We have identified and learned the APIs of the web page and the cookies used. In the case of Giga genie, API could not be obtained for the same reasons as kakao mini. In the case of google home mini, chip readers used in other speakers could not be used because memory chips were different in size. (Absence of chip reader).

Table 12: Successful extraction of data by each acquisition procedure

Acquisition	SK Nugu	Naver Clova	Kakao Mini	Giga Genie	Google home mini
Cloud	O	O	X	X	O
Client	O	O	O	O	O
Network	O	O	O	O	O
Device	O	O	O	O	X
Result	**O**	**O**	**O**	**O**	**O**

The more detailed results of the IoT data acquisition are shown in Table 13. This table is made by summarizing the IoT Data Acquisition procedure presented in chapter 3. If data can be collected, an O mark has been entered. Since this experiment was based on the basic assumption (basic scenario) from the beginning, we excluded Cloud 3 and 4. The procedures that could not acquire any data were Cloud 5, Cloud 6-A, Client 1, and Device 1, Device 2. The reasons are as follows: Cloud 5 and Cloud 6-A are targeted at security vulnerabilities in the cloud. IoT can use APIs that do not use credentials because IoT Product is released in various forms without existing standards or developed by users themselves. That is, not all client apps can have the same security level. while analyzing data from a smartphone to find credential information, we found the ID and password used on a website (Figure 33). However, the artificial intelligent speakers tested at this time did not find this vulnerability.

Figure 33: The ID and PW of the website found during client app analytics

Client 1 was limited in this experiment. IoT devices used in the experiment except Google home mini do not use the browser app. In the case of the Google Home Mini, the browser app does not control the Google Home Mini directly, but rather helps to check the activity history of user accounts using the Google Home Mini. In other words, this method would be suitable for IoT devices that use a browser app to allow some data to be processed on the local computer.

The network traffic between the device/app and the cloud obtained from the network acquisition uses TCP, TLSv 1.2, ARP protocols and the data is encrypted. And the capture of the traffic after the occurrence of the crime may be meaningless. However, we can expect to obtain credential information from the traffic that the cloud app or device tries to connect to the cloud by collecting packets. and IoT network traffic continues to communicate with cloud servers even

when it does not give any orders. And the device automatically updates itself.

Device1 is a method for obtaining data from a device using an API. For example, google onhub can receive diagnostic reports from devices using the API when there is a power outage or network problem and it does not work. Device Level APIs can be formal or informal. But we couldn't find device level for the device used in the case studies. Device2 has been applied to Nest Thermostat in 2014. Device rooting through the USB port on the device exterior and controlling the device as user convenience [27], [28]. However, after the firmware update, the rooting through the USB port connection has been blocked as logically. Among the five artificial intelligence speakers, Kakao mini and Giga genie have external connection port such as USB. Kakao mini's USB port was for charging purposes and Giga genie's USB port was for the media file. And it was not possible to access the inside of the device software through the USB port. and the other speakers don't have any port external.

Table 13: Successful extraction of data by each extraction method detail

Acquisition			**SK Nugu**	**Naver Clova**	**Kakao Mini**	**Giga Genie**	**Google home mini**
Cloud	①		O	O	O	O	O
	②		O	O	O	O	O
	③	**Exclude according to the scenario**					
	④						
	⑤		X	X	X	X	X
	⑥	**A**	X	X	X	X	X
		B	O	O	X	X	O
Client	①		X	X	O	X	O
	②		O	O	O	O	O
Network	①		X	X	O	X	O
	②	**A**	O	O	X	X	O
		B	O	O	O	O	O
Device	①		X	X	X	X	X
	②		X	X	X	X	X
	③	**A**	X	X	X	X	X
		B	O	X	X	X	X
		C	O	O	O	O	X

One of the goals of this case study is to measure the amount of data acquired. As shown in

Table 14 below, data acquisition in the cloud and network acquisition cannot be represented by a specific value. Because it depends on the time of network capture and depends on the amount of data being stored in the cloud or the API provided. The data that can be obtained from the client and device acquisition will vary depending on the amount of memory that can be stored, but the maximum amount of data that can be acquired before the acquisition can be estimated.

Table 14: Data from each extraction method

Acquisition	SK Nugu	Naver Clova	Kakao Mini	Giga Genie	Google home mini	NOTE
Cloud	**Command history**	**Command History**	**Login history, Connected service, Connected device,**	-	**Activity History, Voice record**	
Network	APIs, Credential information, Network traffic	APIs, Credential information.	APIs	-	APIs Credential information	Network Traffic of all devices can be collected
Client	Credential information, Device configuration,	Credential information	Credential information, Personal information	**Command History,** Credential information	Credential Information	Acquired Phone image 27.07GB from all cases
Device	Credential information,	Credential information	Credential information, **Voice Response File**	Credential information **Voice Response File**	-	
	Serial-Port 6.94GB Chip-off 6.0GB	Chip-off 6.0GB	Chip-off 3.9GB	Chip-off 4.3GB		

Though this is not the scope of this case study of analyzing the acquired data, we had to analyze the data acquired from the client or device for credential information to access the cloud data. Data analysis shows that we can get credential information from all client apps, and credential information is found in all the data from the device. Files related to voice commands were found in the data of Kakao mini and Giga genie obtained from the device. The Kakao mini had all the audio files that kakao mini responded to when the user commanded, and the Giga genie had some user voice recording files left. Assuming that the data available as potential evidence is command history, In the case SK Nugu, Naver Clova, and Google Home mini, we could get it

from the cloud, and in the case of Kakao mini and Giga genie, and we can get it from the data of client app or device.

4.7 Conclusion of case studies

These case studies demonstrate that data acquisition is possible on all IoT devices following the proposed IoT Data Procedure acquisition procedure. IoT data acquisition should be done on all sides of the Cloud, Network, Client, and Device as much as possible. There are two reasons for this: First, To access cloud data, the investigator needs to use the credentials from the client app or device and the API obtained from the network side. Second, each side may have different data. In the case of the Kakao Mini, there was a recorded audio file in the device data that indicated the command history, and in the case of the Giga genie, we could find some of the user command histories in the client app.

In this case studies, if the client app provided services to the user to view the activity history, we could get the full history through the API. However, if the client app does not provide a history of commands, even if the command history was obtained through the API, it was some of history, but not all. In other words, if the client app provides a service that allows the user to view command history through the cloud, at least command history information can be obtained from the cloud. In this case study, client apps from Naver Clova and Google home mini provided command history to the user through the cloud, and we were able to get the full command history. But for SK Nugu, we got a very limited command history. In other words, if the type of data that the cloud provides to the client app is identified before performing IoT data acquisition, the investigator can prioritize which side of the cloud, client, network, or device to investigate first.

The data obtained from the network acquisition was not easy to analyze as encrypted packets. But one thing we found that even though the user was not using AI speakers, it were exchanging data with the cloud server and even upgrading the firmware automatically, regardless of the user's intentions. This means that data can be distorted or deleted over time. If there is no way to acquire volatile data from an IoT device, or if the data available as evidence is non-volatile, it is more helpful for investigators to enter the crime scene, identify the device, and turn off the connected Internet or power.

CHAPTER 5. CONCLUSION AND DISCUSSION

This paper presents IoT Data Acquisition procedure through testing of actual IoT devices. The IoT investigation procedure model presented so far has been too general without testing, or the tested research has specialized in specific technologies. The reason is that many variable factors such as the variety of IoT devices and new technologies have created challenges. so it could not be generalized as a comprehensive IoT Investigation procedure. However, Due to the interconnection of IoT devices, the development of IoT devices has begun to take a standard form, and the IoT digital forensic case has been generated. By compiling the digital forensics cases of IoT devices, prior research and our research experience, We propose the IoT data acquisition procedure for the smart home. The procedure was tested with AI speakers that serve South Korea. The AI speakers used for the test were SK Nugu, Naver clova, Kakao mini, Giga genie, and Google home mini which provide service to Korea.

This procedure is classified as Cloud acquisition, Client acquisition, Network acquisition, and Device acquisition. If the location and type of data, API required can be known through prior research or actual case before a criminal investigation, the side that has the required data can be executed immediately. if not, the collection procedures for clients, devices, network, and cloud should be carried out as well as possible together. This procedure did not include the analysis step. But in order to get credentials such as Cookie and Token or APIs, the data from the client, the network, and the device must be analyzed to some extent. A case study for testing was acquired from all sides of the Cloud, Client, Network, and Device, assuming no prior information, such as APIs or data locations. The results of the case study were able to acquire data from all devices. And the location and type of data were different from device to device. In other words, the method used to acquire data in each procedure was different. In other words, data acquisition of IoT

devices should be carried out with the possibility of data being present in all aspects of Cloud, Network, Client and Device.

Through these results, we present the IoT Investigation procedure. In this IoT investigation model and procedures, we provide a generic investigation model and procedures that IoT investigators can follow during investigating smart home IoT devices. The procedure is not intended to give detail guidelines for each tasks in the process, however, to highlight the major starting points in investigating IoTs. Moreover, the procedure uses or refers to other already matured procedures in some areas where we think the process overlaps to existing digital forensics procedures. In this procedure, the Acquisition step follows the IoT Data Acquisition presented in Chapter 3. Guidelines on Mobile Device Forensics developed by NIST (NIST Special Publication 800-101 Revision 1) is one of the main existing guidelines we based our model and procedures. The model has 6 process with each process having different subprocess that need to be executed to fully investigate the IoT device.

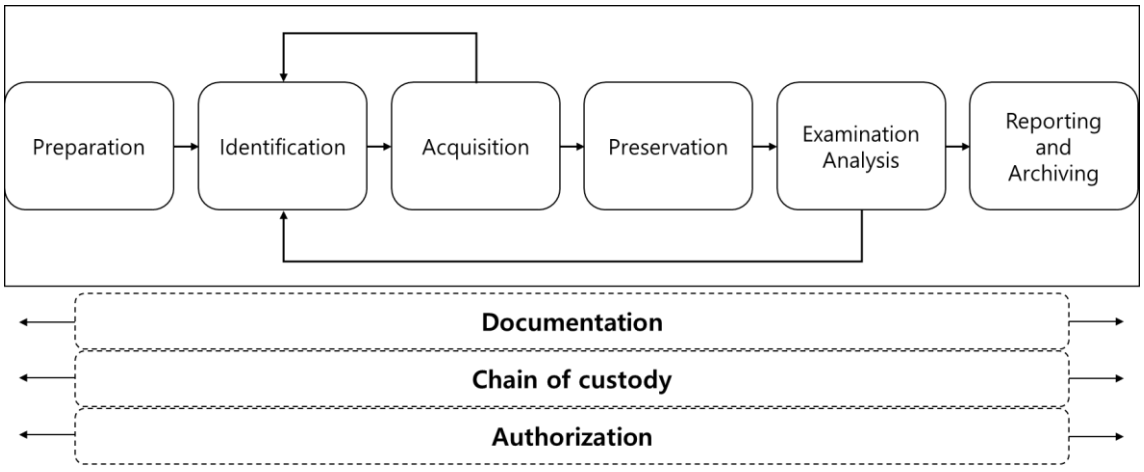

Figure 34: IoT Investigation Procedure

Preparation is a process of being ready to perform digital investigation when the incident is noticed/reported. For IoT investigations, using the available standard operation principles for digital investigation could be helpful. For this investigation procedure, the preparation from Scientific Working Group on Digital Evidence (SWGDE) best practices for digital evidence collection is included as initial point. Preparation in SWGDE is stated as a process that "includes communication between the examiner and investigative team", about the details of the investigation, the nature, and scope of the potential evidence to be acquired, and unique constraints that may impact the acquisition. Also, the need for the examiners to consult appropriate legal counsel if clarification on legal authority is required [34]. The preparation process includes:

- planning
- understanding the case in hand or the context including case evaluation
- preparation detail design on how to conduct the investigation
- allocating resources such as investigation teams
- preparing tools, etc
- securing the crime scene

Note that, some of the IoT to be identified in the crime scene might have not been investigated yet and has not properly tested and verified tools to acquire the data. Therefore, this should be noted at the preparation stage and a strategy should be in place to handle such cases.

Identification is the process of identifying the potential sources of digital evidence that are required to support a claim about a crime [35]. In IoT investigation the process includes identifying the IoT devices, locating them and assessing the value of the device to the specific case to be investigated before performing data extraction. In addition, it should be identified whether the device's information, such as the purpose of the device, the data handled, the type of app is provided, the API identified by prior research, and the tools required, are available. The

main goal in this process is to assess and document the IoTs and integrated devices/services, determine their locations, status and applications/purpose in the house, time and also may be the contribution to the case/crime to be investigated.

Acquisition is the process of extracting evidential data from IoT devices and their cloud storage. This process can be conducted either manually or in an automated way using forensics tools depending on the IoT device and associated cloud storage. As a result, this process depends on the availability of the forensics tools and the capability of the involved investigation specialists. The procedure for this acquisition step is the IoT Data Acquisition procedure presented in this paper.

Preservation is the process of keeping or maintaining the integrity of the extracted data or the evidence/potential evidence data in a way that can be verified at any time. In IoT investigation process, this can be achieved using the standard digital investigation procedures such as: hashing the data, encrypting the data, working on the copy of the data, storing the data separately for each IoTs in specific to the IoT preservation. This process can be also achieved simultaneously with the data collection/acquisition process, examination, and analysis process.

Analysis is the process of determining the forensic value of the extracted data for the specific investigation case in hand. Bulbul et al. in [36] states analysis and examination as a careful observation of the extracted digital evidence to trace user activities and identities without altering the data. The output of this process has significant value to construct evidence for the case in hand. The examination and analysis tasks for digital evidence from IoTs depends on the source of the evidence and the structure of the data, which requires careful analysis and domain knowledge about the investigated IoTs.

Reporting is the process of presenting the investigation findings regarding the case. The presentation includes the findings, the activities performed, the tools and procedures used, investigated IoT devices; to support the hypothesis for the case in hand. In Guidelines on Mobile Device Forensics developed by NIST (NIST Special Publication 800-101 Revision 1)[26]. this process is stated as “the process of preparing a detailed summary of all the steps taken and conclusions reached in the investigation of a case. Reporting depends on maintaining a careful record of all actions and observations, describing the results of tests and examinations, and explaining the inferences drawn from the data. A good report relies on solid documentation, notes, photographs and tool-generated content. Reports of forensic examination results should include all the information necessary to identify the case and its source, outline the test results and findings, and bear the signature of the individual responsible for its contents.”

The features of the proposed IoT Investigation procedure include the process of returning to the Identification during Acquisition and Examination Analysis. Case study results in chapter 4 Device configuration, connected device information, and so on, have been found on the IoT device, which can also be information that can be used to identify unidentified devices, connections, and so on. IoT devices are connected to the Internet and provide services, but other protocols such as Zigbee and Zwave can be connected to other devices. Investigation of IoT devices cannot proceed with a complete identification, as any object can be an IoT device through a network connection. Since there are various shapes and sizes such as ultra-small sensor, implant IoT device, refrigerator, mirror, etc., if any information is found in acuqisition step and analysis step, re-entry of Identification step is required.

This paper describes the procedure for data acquisition in IoT devices in a smart home. This procedure has been tested and validated with five artificial intelligent speakers, which are actual IoT devices. In addition, the IoT investigation procedure was presented using the results of the

IoT Data Acquisition Procedure test. these procedures are the basic guideline for acquiring IoT data to investigators. This will tell you where to start to acquire IoT Data and tell you what the next step is.

REFERENCE

[1] R. Minerva, A. Biru, and D. Rotondi, “Towards a definition of the Internet of Things (IoT),” *IEEE IoT Initiat. white Pap.*, no. 1, p. 86, 2015.

[2] K. Rose, S. Eldridge, and L. Chapin, “The Internet of Things: An Overview - Understanding the Issues and Challenges of a More Connected World,” *Internet Soc.*, no. October, p. 80, 2015.

[3] J. F. Gantz, D. Reinsel, and J. Rydning, “White Paper The U . S . Datasphere : Consumers Flocking to Cloud Sponsored by : Seagate,” no. January, 2019.

[4] Datafair, “IoT Applications | Top 10 Uses of Internet of Things,” *DATAFLAIR*, 2018. [Online]. Available: https://data-flair.training/blogs/iot-applications/.

[5] S. Perumal, N. Md Norwawi, and V. Raman, “Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology,” *2015 5th Int. Conf. Digit. Inf. Process. Commun. ICDIPC 2015*, pp. 19–23, 2015.

[6] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, “Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges,” *Futur. Gener. Comput. Syst.*, vol. 92, no. May 2018, pp. 265–275, 2019.

[7] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, “Internet of Things Forensics: Challenges and Approaches,” *Proc. 9th IEEE Int. Conf. Collab. Comput. Networking, Appl. Work.*, pp. 608–615, 2013.

[8] S. Alabdulsalam, K. Schaefer, T. Kechadi, and N. A. Le, “INTERNET OF THINGS FORENSICS : CHALLENGES,” pp. 1–13, 2017.

[9] M. Harbawi and A. Varol, “An improved digital evidence acquisition model for the Internet of Things forensic I: A theoretical framework,” *2017 5th Int. Symp. Digit. Forensic Secur. ISDFS 2017*, 2017.

[10] P. H. Rughani, “IoT Evidence Acquisition – Issues and Challenges,” vol. 10, no. 5, pp. 1285–1293, 2017.

[11] R. C. Hegarty, D. J. Lamb, and A. Attwood, “Digital Evidence Challenges in the Internet of Things,” *Proc. Tenth Int. Netw. Conf. (INC 2014)*, 2014.

[12] S. Watson and A. Dehghantanha, “Digital forensics: the missing piece of the Internet of Things promise,” *Comput. Fraud Secur.*, 2016.

[13] B. Christian, “Digital Forensics on Small Scale Digital Devices,” 2009.

[14] H. Chung, J. Park, and S. Lee, “Digital forensic approaches for Amazon Alexa ecosystem,” *Digit. Investig.*, vol. 22, pp. S15–S25, Aug. 2017.

[15] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, “Chapter 2 CLOUD FORENSICS,” *IFIP Adv. Inf. Commun. Technol. B. Ser.*, 2011.

[16] J. S. Hale, “Amazon Cloud Drive forensic analysis,” *Digit. Investig.*, vol. 10, no. 3, pp. 259–265, 2013.

[17] H. Chung, J. Park, S. Lee, and C. Kang, “Digital forensic investigation of cloud storage services,” *Digit. Investig.*, vol. 9, no. 2, pp. 81–95, 2012.

[18] B. Martini and K. K. R. Choo, “Cloud storage forensics: OwnCloud as a case study,”

Digit. Investig., vol. 10, no. 4, pp. 287–299, 2013.

[19] V. Roussev and S. McCulley, “Forensic analysis of cloud-native artifacts,” *Digit. Investig.*, vol. 16, pp. S104–S113, 2016.

[20] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng, and Y. Xiao, “Network forensics analysis using Wireshark,” *Int. J. Secur. Networks*, vol. 10, no. 2, p. 91, 2015.

[21] S. Pappas, “Investigation of JTAG and ISP Techniques for Forensic Procedures,” 2017.

[22] I. Clinton *et al.*, “A Survey of Various Methods for Analyzing the Amazon Echo,” 2016.

[23] M. Barnes, “Alexa-are-you-listening,” *MWR LABS*, 2017. [Online]. Available: https://labs.mwrinfosecurity.com/blog/alexa-are-you-listening/.

[24] V. R. Kebande and I. Ray, “A generic digital forensic investigation framework for Internet of Things (IoT),” in *Proceedings - 2016 IEEE 4th International Conference on Future Internet of Things and Cloud, FiCloud 2016*, 2016.

[25] E. Akbal, F. Güneş, and A. Akbal, “Digital Forensic Analyses of Web Browser Records,” *J. Softw.*, vol. 11, no. 7, pp. 631–637, 2016.

[26] R. Ayers, W. Jansen, and S. Brothers, “Guidelines on mobile device forensics (NIST Special Publication 800-101 Revision 1),” *NIST Spec. Publ.*, vol. 1, no. 1, p. 85, 2014.

[27] S. Darlene, “Black Hat: Nest thermostat turned into a smart spy in 15 seconds,” *Computerworld*, 2014. [Online]. Available: https://www.computerworld.com/article/2476599/black-hat-nest-thermostat-turned-into-a-smart-spy-in-15-seconds.html.

[28] Brian Benchoff, “ROOTING THE NEST THERMOSTAT,” *HACKADAY*, 2014. [Online]. Available: https://hackaday.com/2014/06/24/rooting-the-nest-thermostat/.

[29] SWGDE, “SWGDE Best Practices for Chip-Off,” vol. 0, pp. 1–11, 2016.

[30] Y. Hiroto, K. Chong-Min, and L. Yongpan, *Smart Sensors at the IoT Frontier*. .

[31] Nasmedia, “DIGITAL MEDIA & MARKETING TREND FORECASTING,” 2019.

[32] SKT, “T Developer,” 2017. [Online]. Available: https://developers.sktelecom.com/community/pr/view/?ntcStcId=20160201171538.

[33] A. Counter, “검색엔진 유입률 분석(2018년 3/4분기),” 2018.

[34] S. B. Practices and C. Forensics, “Scientific Working Group on Digital Evidence Scientific Working Group on Digital Evidence,” vol. 1, pp. 1–12, 2014.

[35] M. Khan, S. Din, S. Jabbar, M. Gohar, H. Ghayvat, and S. C. Mukhopadhyay, “Context-aware low power intelligent SmartHome based on the Internet of things,” *Comput. Electr. Eng.*, vol. 52, pp. 208–222, 2016.

[36] H. I. Bulbul, H. G. Yavuzcan, and M. Ozel, “Digital forensics: An analytical crime scene procedure model (ACSPM),” *Forensic Sci. Int.*, vol. 233, no. 1–3, pp. 244–256, Dec. 2013.

A Study on Internet of Things (IoT) Forensic

2019
Master's Degree
Jang, Subong
Department of International Studies
Advisors: Prof. Joshua I. James, Prof. Jang, Yoonsik

There are previous studies about digital forensics in terms of the Internet of Things (IoT) environment, and cases using IoT device data for crime investigations. However, the acquisition and analysis of data in the IoT environment are still a challenge for digital forensic investigators. In addition, there are no accepted practical and comprehensive digital forensic procedures for investigators and law enforcement agencies to perform digital forensic investigations on IoT-based environments.

This work proposes a new model for IoT Forensic specifically for the Data Acquisition procedure from the IoT ecosystem. The model is tested using experiments conducted on IoT devices. The experiment includes the research aspect of investigating the actual IoT devices in order to provide a complete picture of the model. The experiment was divided into Cloud, Network, Client (PC, Mobile) and Device/hub side for each IoT device.

The results from the experiments showed that data can be extracted from each category of cloud, network, client, and device, and that the data should be collected as soon as possible with the related devices and collected as much as possible. This is

because the data available in the device and in the specified categories can vary depending on the storage and processing capabilities. For example, the device side may have data that the cloud side does not have. The data can be key evidence for a crime scene. And the cloud side includes the more complete update and historical data, while the client and device side include cache data that may be incomplete, outdated, or partially overwritten.

Through these test results, this paper also presents the IoT investigation procedure. In order to proceed with data collection from all aspects of IoT ecosystem, it is necessary to analyze data obtained from the client side. Information such as device configuration information and connected devices can be obtained by acquiring and analyzing them. By checking the information of the other connected IoT devices, the investigation can go through the identification step again. The proposed procedure will serve as a guideline from where to start to investigate IoT devices and what the next step is. It will also facilitate investigators and researchers in the digital forensics field to facilitate the data acquisition process and develop data collection tools. This procedure should be tested and verified against a variety of IoT devices until a comprehensive procedure for IoT data acquisition.

Keyword: Internet of things forensic, IoT Forensic, IoT data acquisition, Cloud Acquisition, Network Acquisition, Client-side Acquisition, Device Acquisition.

사물인터넷 포렌식에 관한 연구

2019
석사학위논문
장수봉
국제학과
지도교수: Joshua I. James, 장윤식

Internet of Things(IoT) 환경에서 Digital Forensic에 위한 많은 연구가 이루어지고 있으며, 실제 IoT 장치 분석 사례 및 범죄 사건에 IoT 장치의 데이터를 이용한 사례들이 발생하고 있다. 하지만 IoT 기반의 인프라에서 Digital forensic investigation을 수행하는데 수사관들이 쓸 수 있는 실용적이고 포괄적인 절차는 완성되지 않았으며, IoT 환경에서 데이터를 획득, 분석하여 증거로서 채택하는 것은 법원과 Digital forensic 수사관에게는 아직까지 도전 과제이다. 본 논문은 IoT 장치를 분석한 경험과 IoT 분석 사례 연구를 기반으로 IoT Data Acquisition 절차를 제안하고, 실제 IoT 장치(인공지능스피커)를 이용한 실험을 통해 제시된 절차를 검증하였다. 실험은 IoT 장치마다 Cloud, Network, Client(PC, Mobile), Device 측으로 나누어 Data Acquisition이 수행되었으며, 실험 목적은 IoT 장치를 이용하여 제시된 절차를 따라 Data Acquisition 수행하여 데이터 획득가능 여부와 얼마나 많은 데이터를 획득할 수 있는지 확인한다. 실험결과는 각 측면마다 데이터를 추출이 가능하였으며, 데이터 습득은 관련된 장치들과 가능한한 함께 수행되어져야 함을 나타냈다. 그 이유는 첫째, 각 측면마다 얻을 수 있는 데이터가 다를 수 있기 때문이다. 예를 들어, 클라우드에는 저장되어 있지 않은 데이터가 장치에 저장되어 있을

수 있다. 둘째, 클라우드 데이터에 접근가능한 자격증명정보는 클라우드 측과 디바이스 측에서 모두 얻을 수 있었다. 이것은 만약 클라이언트 측의 데이터 획득이 불가 할 때는 디바이스에서 얻을 수 있음을 나타냈다. 이러한 테스트 결과를 통하여 본 논문은 IoT 수사 절차도 함께 제시한다. IoT 생태계의 모든 측면에서 데이터 획득을 진행하기 위해서는 클라이언트 측에서 얻은 데이터를 분석할 필요가 있으며, 획득과 분석을 하면서 장치구성정보, 연결된 장치와 같은 정보를 얻을 수 있다. 이처럼 연결된 다른 IoT 장치의 정보를 확인함으로써, IoT 수사는 다시 식별 단계를 거쳐야 할 수 있다. 제시된 절차는 IoT 장치를 조사하기 위해 어디서부터 시작하고, 다음단계는 무엇인지 가이드라인의 역할을 할 것이다. 또한 디지털 포렌식 분야의 수사관과 연구자에게 데이터 획득 프로세스를 용이하게 하며, 데이터 획득 도구를 개발하는 데 기여할 수 있다.

주제어: 사물인터넷, Internet of Things, IoT 디지털 포렌식, 클라우드 데이터 추출, 네트워크 데이터 추출, 클라이언트측 데이터 추출, 디바이스 데이터 추출.