ISSN(Print) : 1976-5304, ISSN(Online) : 2713-931X
https://kdfs.jams.or.kr, DOI : 10.22798/kdfs.2023.17.2.45

원격 제어용 어플리케이션에서의 아티팩트 수집 및 분석

박 현 재*, 손 태 식**
아주대학교 사이버보안학과 (학부생)*, (교수)**

Artifact Collection and Analysis in Remote Control Applications

Hyun-Jae Park*, Tae-Sik Shon**
Dept. of Cybersecurity, Ajou University* (Undergraduate Student)*, (Professor)**

요 약

코로나-19 대유행 및 원격 근무의 증가로 원격 제어 소프트웨어가 기업 및 개인 간에 물리적인 액세스가 불가능한 경우에 협업과 물리적 중요한 역할을 하고 있다. 그러나 최근 범죄 조직은 메신저를 활용하여 피해자에게 원격 제어 앱을 설치하도록 유도하고, 이를 통해 다양한 개인정보를 탈취하며 금융 사기를 일으키는 사례가 증가하고 있다. 이에 따라 본 연구는 TeamViewer, Anydesk, AirDroid 등의 원격 제어 어플리케이션을 대상으로 디바이스에 저장된 아티팩트의 수집 및 분석을 통해 범죄에 활용된 아티팩트를 분석하고, 범인에 대한 추측 가능한 로그인 계정 정보 또는 닉네임 설정이 이루어진 경우에 수사 활용 방안을 제시한다. 따라서 본 연구는 앞서 확인한 한계를 토대로 후속 연구의 필요성을 제기한다.

주제어 : 디지털 포렌식, 안드로이드 포렌식, 원격 제어, 보이스피싱, 팀뷰어, 디지털 증거 분석

ABSTRACT

Due to the COVID-19 pandemic and the rise of remote work, remote control software is playing a crucial role in facilitating collaboration between businesses and individuals when physical access is not possible. However, there has been an increase in criminal organizations exploiting messaging platforms to manipulate victims into installing remote control applications, leading to the theft of various personal information and the perpetration of financial fraud. In response to this issue, this study focuses on remote control applications such as TeamViewer, Anydesk, and AirDroid. It involves the collection and analysis of artifacts stored on devices to understand those used in crimes, and proposes investigative methods that can be employed when it is possible to hypothesize about the perpetrator based on login account information or nickname settings found in the artifacts. Recognizing the limitations identified in this study, it underscores the necessity for further research in this area.

KeyWords : Digital forensics, Android forensics, Remote Control, Voice Phishing, TeamViewer, Digital Evidence Analysis

Ⅰ. 서 론

원격 제어용 소프트웨어는 컴퓨터나 스마트폰에 물리적인 액세스가 불가능할 경우 개인이 파트너의 기기에 액세스 할 수 있도록 해주는 소프트웨어이다. 코로나-19의 대유행으로 원격 근무로의 전례 없는 전환으로 많은 사람들이 직접 회사에 출근하지 않고도 집의 디바이스로 작업하고 있으며 이에 따라 협업 및 시스템을 유지하고 보안을 유지하기 위해서 원격 제어 소프트웨어 등에 의존하고 있다[1]. 전통적인 파일 전송 프로토콜(FTP)와는 달리 파일을 전송하는 것뿐 만 아니라 호스팅 디바이스에 설치되어 있는 어플리케이션을 열람하는 것에 더불어 어플리케이션을 직접 조작하거나 시스템 값의 설정을 변경하는 등의 구체적인 행위가 가능하다.

※ 본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음(2022-0-01077)
- Received 14 January 2024, Revised 6 February 2024, Accepted 29 March 2023
- 제1저자(First Author) : Hyunjae Park (Email : ing07006@ajou.ac.kr)
- 교신저자(Corresponding Author) : Taeshik Shon (Email : tsshon@ajou.ac.kr)

원격제어 기능이 실행 중일 때는 자신의 디바이스 화면이 상대방에게 그대로 전송되며, 마우스, 키보드, 터치 등 각종 입력 기능 역시 상대가 제어할 수 있다[2]. 따라서 디바이스나 소프트웨어 서비스를 제공하는 업체에서 고객에게 원격으로 지원하기 위해서 하지만 만약 그 '상대방'이 범죄조직이라면 자신의 스마트폰을 마치 범죄 조직의 물건인 것처럼 물리적으로 먼 거리에서도 문자메시지, 주소록, 사진 및 동영상 등을 유출하고, 알 수 없는 악성 앱을 설치하는 등 기기의 보안 상태를 취약하게 만드는 것도 가능하다.

최근 범죄 조직이 원격제어 앱인 '팀뷰어'를 이용해 개인정보와 금융정보를 탈취하고, 이를 악용해 금융사고를 일으키는 사례가 증가하고 있다[3]. 이를 통해 범죄조직은 신규 스마트폰 개통, 휴대전화 번호 변경, 인터넷 은행 및 증권 계좌 개설, 신용대출 등에 이용하고 있다. A씨의 사례를 통해 이러한 범죄의 진행 과정을 상세히 보여준다[4]. A씨는 딸을 사칭한 범인에게 개인 정보를 제공하고, '팀뷰어' 설치를 요청받아 설치했다. 범인은 이를 통해 A씨의 휴대폰을 원격으로 제어하고, 공인인증서와 디지털 OTP 재발급, 신용카드 대출 등 금융사기를 행했다. 또한, 광주와 대전 등지에서도 '팀뷰어'를 이용한 비슷한 사례가 발생하기도 했다.

이렇듯 원격 제어 어플리케이션은 범죄자에게 제어권이 넘어간다면 금융 사기 피해로 이어질 수 있다[5]. 사기 피해액이 점점 커짐에 따라 중요도가 높지만 현재 원격 제어 어플리케이션에 대한 포렌식 연구는 매우 저조하다. 따라서 원격 제어 어플리케이션을 통해 금융 사기 범죄가 일어났을 때, 범죄자가 피해자의 디바이스에 어떤 행위들을 했는지를 연구를 통해 파악해야 한다. 연구를 통해 해당 금융 피해 사례에서 원격 앱을 통해 범인이 어떤 디바이스를 가지고 원격 제어를 수행했는지, 어떤 파일을 전송하고 원격으로 이루어진 이벤트 등을 식별하였는지 추가적으로 탈취한 정보가 어떤 것인지를 알게 된다면 실제 사건 수사에서 범인에게 어떤 정보가 넘어가게 되었는지를 알게 되어 수사에 도움을 줄 수 있을 것이다. 따라서 현재 널리 사용되고 있는 원격 제어 어플리케이션 팀뷰어, Anydesk, AirDroid 세 가지의 어플리케이션에서의 디바이스에 저장된 아티팩트의 수집 및 분석을 통해 범죄에 사용된 아티팩트를 이해하고, 이러한 아티팩트를 어떻게 수사에 활용할 수 있는지에 대한 방안을 연구해보아야 한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존의 원격 제어 소프트웨어 아티팩트 분석에 관련된 선행 연구를 소개한다. Ⅲ장에서는 본 논문에서 수행한 아티팩트 수집 방법과 이를 수행한 환경에 대해서 소개한다. Ⅳ장에서는 각 어플리케이션 별로 수집된 아티팩트에 대한 정보와 경로에 대해 소개하고 각각의 주요한 특징을 비교한다. Ⅴ장에서는 각 어플리케이션에서 수집한 주요 아티팩트를 정리하고 앞서 소개했던 아티팩트로 두 개의 시나리오를 작성하여 수사에 활용할 방안을 제시한다. 마지막으로 Ⅵ장에서는 원격 제어 어플리케이션 아티팩트 조사에 대한 결론과 향후 연구 발전 방향을 제시한다.

Ⅱ. 관련 연구

Colby Lahaie[6]는 Windows에서 피해자가 TeamViewer에 의해서 접근 당하고 있을 때의 상황을 Windows 환경에서의 TeamViewer가 실행되는 동안의 포렌식 연구를 진행하였다. 이들은 TeamViewer가 남긴 로그 파일에는 가장 많은 데이터가 들어 있으며, 세션 동안 무슨 일이 일어났는지, 모든 파일 전송 및 세션 참가자에 대한 정보를 포함하여 대부분을 보여주었다. "TeamViewer8_Logfile.log"라는 이름으로 불리는 로그파일은 모두 TeamViewer 설치 폴더에 저장되며, 기본적으로 TeamViewer 버전 8의 경우 "C:\Program Files (x86)\TeamViewer\Version8"이었다. 하지만 이는 윈도우 OS 기반으로 한 TeamViewer의 설치 파일에 대한 분석만 존재하였고, 실질적으로 피싱 범죄에서 사용되는 어플리케이션 형태에 대한 연구는 진행되지 않았기 때문에 이에 대한 연구가 필요하다.

Manson Jonathan[7]는 중요성과 보급도에도 불구하고 원격 데스크톱 소프트웨어에 대한 포렌식 연구는 최소인 현재 상황에 대해 강조하고, TeamViewer가 이러한 소프트웨어가 포렌식적으로 가치 있는 것을 탐색하는 중요한 시작점으로 보고 적절한 증거를 통해 조사관이 어떤 기계가 원격 제어를 수행했거나 제어되었는지, 파일을 전송하고 원격으로 재부팅된 이벤트 등을 식별할 수 있다는 것을 보여주었다. 최종 결과물로 Autopsy용 Python 모듈을 게시하여 조사관이 TeamViewer의 주요 아티팩트를 자동으로 찾고 처리하고 시각화할 수 있도록 하는 연구를 진행하였다.

위의 연구들은 모두 PC 환경, 그 중에서도 윈도우 운영체제에서의 TeamViewer 환경에 대한 조사가 이루어졌다. 안드로이드용 TeamViewer 애플리케이션의 경우 로그가 저장되는 경로가 다르고, 윈도우즈에서 확인 가능한 작업 프로세스 관리자 목록 에서는 안드로이드에서는 프로세스 목록(process list)를 확인하는 것을 통해 해당 애플리케이션이 내 단말기에 작동하고 있다는 것을 알 수 있다. 따라서 선행 연구들에서는 안드로이

드 환경에서의 아티팩트를 조사하지 않았기 때문에 안드로이드 파일 시스템 내에 어떠한 경로에 로그가 저장되는 지를 중점적으로 연구를 진행할 예정이다.

Ⅲ. 분석 대상의 기능 및 특징과 분석 환경

3.1. 분석 대상 어플리케이션의 아티팩트 수집 과정

실험은 각 어플리케이션을 통해 해당 디바이스를 직접 원격 제어를 하고 디바이스와 PC를 유선 연결하여 adb로 디바이스 내 어플리케이션의 /data/data 파일을 이미지 파일로 추출하여 분석하였다. 분석에 사용된 어플리케이션은 2023년 12월을 기준으로 원격 제어 어플리케이션 중 가장 사용량이 많은 3종에 대해 분석하였다. 각 어플리케이션은 다른 디바이스를 통해 안드로이드 기기를 원격 제어를 가능하게 해주는 기능을 포함하는 공통적인 특징을 가지고 있다. [표 1]에는 사용한 어플리케이션의 종류와 버전과 다운로드 수 정보와 실시간으로 제어받는 기기의 화면을 제어하는 기기로 관찰하는 하는 기능을 제공하는 지에 대한 여부를 확인할 수 있다.

〈Table 1〉 List of applications of remote control application

Applications	Version	Donwloads	Providing real-time control of device screen
TeamViewer - QuickSupport	15.48.352	50 Million more	O
AnyDesk	7.0.0	50 Million more	O
AirDroid	4.3.2.0	50 Million more	X

3.2. 실험 환경

조사에 사용한 스마트폰은 Magisk[8]를 활용해 루팅하고 busybox[9]를 설치해 휴대폰 내 kernel을 사용하도록 하였다. SM-G998N(SAMGSUNG Galaxy s21 Ultra)이며, Android 12 버전, One UI 버전 4.1에서 진행하였다. 그리고 제어를 하는데 사용한 기기의 OS는 MacOS Ventura를 사용하였고, 어플리케이션 데이터를 분석하기 위해서 Apktool[10]을 활용하여 디패키징을 진행하였고 내부 데이터 분석을 위해서는 HxD를 활용하였다. 그리고 xml 코드 데이터를 열람하기 위해서 Xcode Version 14.3.1을 사용하였다. 제어 받는 기기에서 앱 데이터 이미지를 추출하기 위해서는 루트 권한이 필요하므로 TWRP[11], 삼성 오딘, Magisk를 이용하여 기기의 루팅을 진행하였다. 실험 환경에 대한 모든 정보는 [표 2]와 같다.

〈Table 2〉 Experiment Environment

	Name and Version
Controlled Device / OS	Samsung Galaxy s21 Ultra Android 12
Controlling Device / OS	Apple MacBook M1 Pro Ventura 13.1(22C65)
Data backup	Sansung Odin3 ver 3.14.4 Magisk ver 26.1
Analysis Tool (Viewer)	SQLite Viewer Web App available for web Hex fiend Version 2.15 (1653948392) Xcode Version 14.3.1 (14E300c)

Ⅳ. 아티팩트 수집 및 분석

4.1. TeamViewer

〈Table 3〉 Android TeamViewer Controlling device and Controlled device information

	TeamViewer ID	Version
Controlling Device	1 341 003 884	MacOS 15.48.4 (4486e586978)
Controlled Device	1 341 626 180	Android 15.48.352 QS(QuickSupport)

TeamViewer의 경우 제어를 받는 기기는 TeamViewer의 여러 서비스 중에서 "QuickSupport"를 설치하여야 한다. 따라서 어플리케이션의 패키지명 역시도 QuickSupport로 지정되어 있다. 따라서 앞으로는 TeamViewer 대신 QuickSupport로 지칭하도록 할 것이다. 〔표 3〕에는 제어 받는 기기에서 원격 제어에 사용되는 팀뷰어 ID와 설치된 TeamViewer의 버전 정보에 관한 내용이 담겨있다. QuickSupport의 파일 저장 위치는 /data/data/com.teamviewer.quicksupport.market 이다. 해당 경로 내의 파일들 중에서 유의미한 정보를 찾아보았다. 〔표 3〕은 해당 경로에서 찾은 유의미한 파일들의 정보를 담은 표이다.

〈Table 4〉 Android TeamViewer QuickSupport application data storage path and contents

Path	File Name	File Content
data/com.teamviewer.quicksupport.market/sharedprefs	com.teamviewer.quicksupport.market_preferences.xml	원격 제어 당시의 QuickSupport 어플리케이션에서 요청한 권한 등
data/com.teamviewer.quicksupport.market/files	client.conf	유저의 SUID 정보
	TVLog.html	원격 제어의 시작 단계부터 종료까지 TeamViewer 작업 이벤트 로그

첫 번째로 data/com.teamviewer.quicksupport.market/sharedprefs에 위치한 com.teamviewer.quicksupport.market_preferences.xml 파일인 〔그림 1〕에서 Boolean Type으로 정의된 TeamViewer에서 안드로이드 운영체제에 요청한 권한에 대한 다양한 정보를 얻을 수 있었다. 해당 파일의 내용은 요청 권한에 대한 상세한 설명은 〔표 4〕에서 확인할 수 있다.

com.teamviewer.quicksupport.market_preferences.xml — Edited

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <int name="BUILD_VERSION" value="352" />
    <boolean name="KEY_OVERLAY_PERMISSION_SHOW_SNACKBAR" value="true" />
    <boolean name="ENABLE_BLE" value="false" />
    <boolean name="KEY_SHOULD_LAUNCH_INTRO_THIS_TIME" value="false" />
    <boolean name="IS_FIRST_START_EVER" value="false" />
    <boolean name="useUDP" value="true" />
    <string name="PREFERRED_RESOLUTION">DontChange</string>
    <int name="INPUT_METHOD_INT" value="0" />
    <string name="LAST_VERSION_PROMPTED_FOR_REVIEW">15.48.352</string>
    <string name="VERSION_STRING">15.48.352 QS</string>
    <boolean name="KEY_EULA_ACCEPTED" value="true" />
    <boolean name="VERBOSE_LOGGING" value="false" />
    <boolean name="KEY_SHOW_NOTIFICATION_DISABLED_WARNING" value="true" />
</map>
```

〈Figure 1〉 com.teamviewer.quicksupport.market_preferences.xml file content

〈Table 5〉 Android TeamViewer QuickSupport application permission contents

Path	contents
KEY_OVERLAY_PERMISSION_SHOW_SNACKBAR	원격 제어를 위해 TeamViewer QuickSupport 앱이 다른 앱 위에 표시될 수 있도록 하는 권한
ENABLE_BLE	Bluetooth 탐색을 활성화하여 주변 기기가 이 기기를 찾을 수 있도록 하는 권한
KEY_SHOULD_LAUNCH_INTRO_THIS_TIME	Intro를 이번 실행에 실행하도록 하는 권한
IS_FIRST_START_EVER	첫 원격제어 연결임을 표시하도록 해주는 권한
useUDP	최상의 성능을 위해 UDP를 사용해서 원격 제어 연결을 수립하도록 해주는 권한
KEY_EULA_ACCEPTED	End-User License Agreement를 동의한 사용자를 표시 하도록 해주는 권한
VERBOSE_LOGGING	고급 로깅을 허용하는 지에 대한 권한
KEY_SHOW_NOTIFICATION_DISABLED_WARNING	경고 알림을 Disabled 하는 권한

두 번째로 data/com.teamviewer.quicksupport.market/files에 위치한 client.conf 파일에서는 TeamViewer User Settings에 대한 정보를 얻을 수 있다. 하지만 해당 정보는 Mac용 binary 열람 파일인 hex fiend로 열람한 결과 SUID에 대한 hex 값을 얻을 수 있다. 얻은 hex 값의 형태는 〔Figure 2〕와 〔Figure 3〕에서 확인이 가능하다.

```
1 TeamViewer User Settings
2 # It is not recommended to edit this file manually
3
4
5 [bin  ] SUID = 5b02a7efbd1343e3aa5ac9379b6590fe
6
```

〈Figure 2〉 client.conf file contents

client.conf

0	5465616D	56696577	65722055	73657220	53657474	696E6773	0A232049	74206973	TeamViewer User Settings # It is
32	206E6F74	20726563	6F6D6D65	6E646564	20746F20	65646974	20746869	73206669	not recommended to edit this fi
64	6C65206D	616E7561	6C6C790A	0A0A5B62	696E2020	5D205355	4944203D	20356230	le manually [bin] SUID = 5b0
96	32613765	66626431	33343365	33616135	61633933	37396236	35393066	650A	2a7efbd1343e3aa5ac9379b6590fe

〈Figure 3〉 client.conf file contents from Hex Fiend

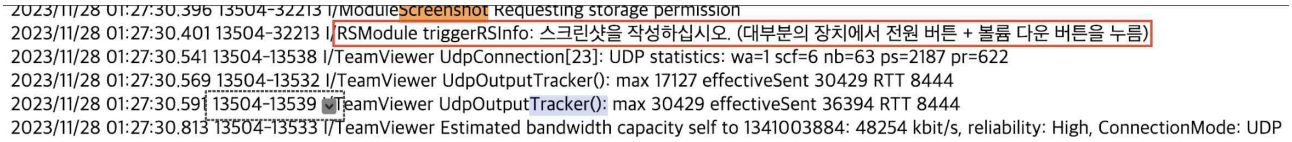
마지막으로 같은 경로 내의 TVLog.html 파일에서는 원격 제어가 이루어질 당시에 작업 내용에 해당하는 로그를 확인할 수 있었다. 해당 로그에는 원격 제어자와의 연결 시작부터 이루어진 활동들이 나열되어 있다. 특히 CPersistentParticipantManager의 문자열로 태깅된 항목을 보면 Controller Device에서 사용한 계정 사용자 명에 대한 정보를 얻을 수 있다.

```
2023/11/24 23:55:46.400 13504-13535 I/TeamViewer CPersistentParticipantManager::AddParticipant: [1341003884,169485818] type=6 name=Park Hyun Jae
2023/11/24 23:55:46.400 13504-13535 I/TeamViewer CPersistentParticipantManager::AddParticipant: [1341626180,1918437688] type=3 name=samsung_SM-G998N_unknown
2023/11/24 23:55:46.401 13504-13535 I/TeamViewer CParticipantManagerBase participant samsung_SM-G998N_unknown (ID [1341626180,1918437688]) was added with the role 3
2023/11/24 23:55:46.415 13504-13504 I/ActivityManager fragmentStarted o.ec2
2023/11/24 23:55:46.492 13504-13539 I/TeamViewer CParticipantManagerBase InteractionDefaults arrived : CInteractionDefaults = (0) [ 0,2,0,0,2,0,0]
2023/11/24 23:55:46.492 13504-13539 I/TeamViewer CParticipantManagerBase participant Park Hyun Jae (ID [1341003884,169485818]) was added with the role 6
2023/11/24 23:55:46.515 13504-13535 W/TeamViewer CStreamManager[10]::IncomingStreamCommand(): anticipating streamID=2!
2023/11/24 23:55:46.516 13504-13535 I/TeamViewer UDPv4: sending pings...: (*)
2023/11/24 23:55:46.517 13504-13533 I/TeamViewer CParticipantManagerBase participant samsung_SM-G998N_unknown (ID [1341626180,1918437688]) was added with the role 3
```

〈Figure 4〉 Controller Account name information from TVLog.html file

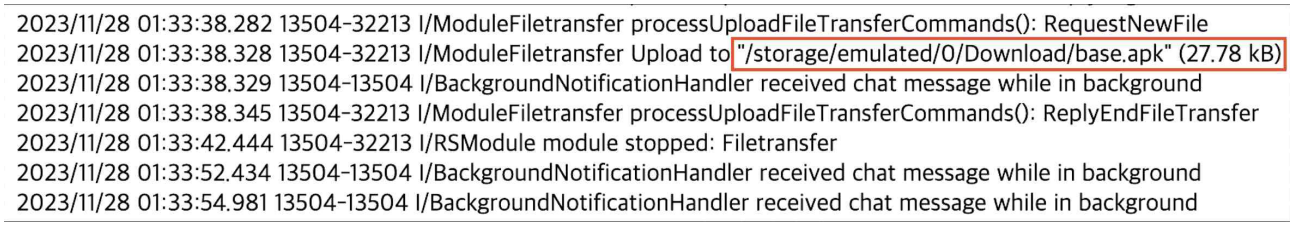
또한, 〔표 3〕에서 확인한 Controller의 계정 정보 중에서 원격 접속에 필요한 ID 정보를 TVLog.html 파

일에서도 확인이 가능하다. [1341003884,169485818] 문자열에서 왼쪽 숫자 문자열은 Controller의 ID 정보에 해당한다. 이를 통해 원격 제어가 이루어진 뒤에도 원격 제어를 받은 유저는 어떤 계정으로부터 제어를 받았는 지에 대해 특정짓는 것이 가능하다. 만일, 원격 제어가 범죄에 활용되어 피해를 입은 상황이라면 해당 아티팩트를 통해 TeamViewer 측에 해당 계정 정보를 가진 사용자에 대해 수사에 협조를 요청하는 식으로 수사에 활용될 수 있다.



〈Figure 5〉 Screenshot Trigging artifacts from TVLog.html file

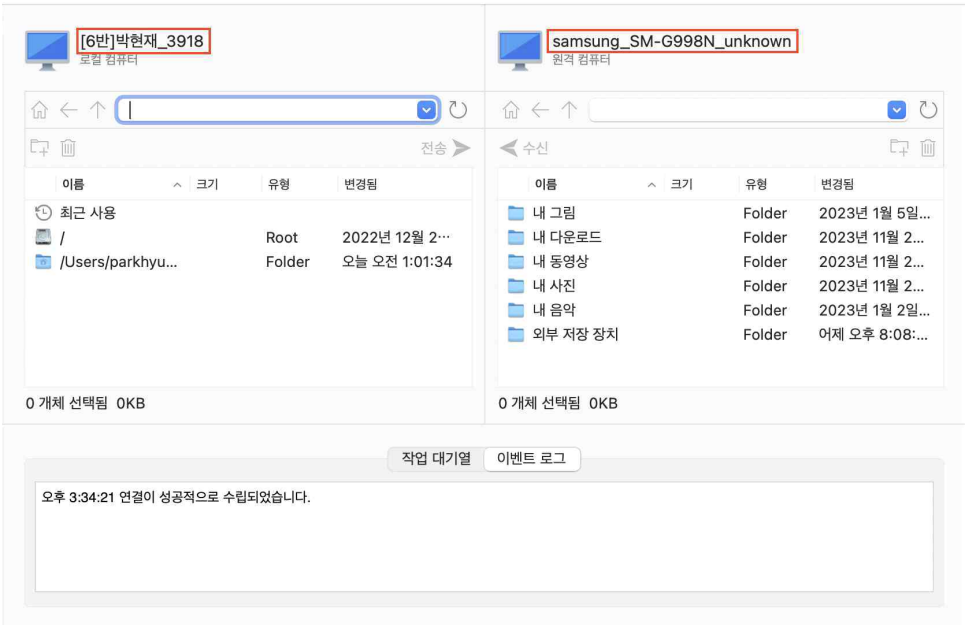
[Figure 5]는 TeamViewer의 내부 앱 데이터 중 files 경로에서 얻을 수 있는 작업 로그에 해당하는 아티팩트이다. 해당 로그 파일에서 추가적으로 활용될 수 있는 아티팩트 정보는 제어자의 스크린샷 요청이다. TeamViewer 원격 제어가 이루어지는 상황에서는 제어자가 제어를 받는 사용자의 화면 캡쳐가 필요한 경우 사용자에게 직접 스크린샷을 촬영을 요청하도록 하는 기능이 존재한다. 이 때 제어자가 스크린샷 요청을 보내는 경우 해당 로그 파일에 [Figure 5]와 같이 스크린샷 요청에 관한 로그가 남는다. 이 때 스크린샷은 제어받는 사용자의 디바이스에 저장됨과 동시에 제어자에게도 동일한 스크린샷 이미지가 넘어가게 된다. 만일, 원격 제어가 범죄에 활용되어 제어자의 민감 정보가 포함된 화면을 스크린샷 하도록 유도한 상황이었다면 제어받은 사용자의 디바이스에 남겨진 스크린샷 이미지의 촬영 시각과 해당 로그에서 제어자가 요청한 스크린샷 요청의 시간을 대조하여 어떤 종류의 민감 정보가 범인에게 넘어가게 되었는 지에 대해서 수사 방향성을 잡는 데에 도움을 얻을 수 있다.



〈Figure 6〉 File Transfer artifacts from TVLog.html file

해당 로그 파일에서 추가적으로 활용될 수 있는 아티팩트 정보는 파일 전송 기록이다. TeamViewer 원격 제어가 이뤄지는 상황에서는 제어자가 제어 받는 기기에 해당 하는 기기에 제약 없이 파일 전송이 가능하다. [Figure 7]에는 원격 제어가 이루어지고 있을 시에 제어자(좌측)와 제어 받는 사용자(우측)의 파일 시스템 목록을 확인한 모습이다. 해당 기능을 활용하면 제어자는 제어 받는 사용자의 백그라운드 환경에서 파일을 전송할 수 있다. 만일, 원격 제어가 범죄에 활용되어 제어자가 제어 받는 사용자의 디바이스에 악성 어플리케이션 파일을 전송하고, 사용자에게 이를 설치하도록 유도했다면 해당 파일을 제어자가 제공했다는 아티팩트가 필요할 것이다.

이 때 활용할 수 있는 아티팩트는 작업로그 파일에 남아있다. 함께 저장되는 태그로는 ModuleFiletransfer로 해당 태그로 저장된 로그의 상세 내역을 보면 제어자가 제어 받는 사용자의 디바이스의 어떤 경로에 어떤 파일을 전송했는 지를 확인하는 것 가능하다. 상세 로그는 [Figure 5]와 [Figure 6]에서 확인이 가능하다. [Figure 6]의 경우에는 제어 받는 사람의 디바이스에서 "/storage/emulated/0/Download" 경로에 "base.apk" 파일을 전송한 사실을 확인할 수 있다. 추후 해당 어플리케이션이 추가적인 범죄에도 활용될 가능성이 있으므로 해당 아티팩트를 통해 전송된 파일을 추적하는 데에 큰 도움을 얻을 수 있다.



〈Figure 7〉 TeamViewer File Transfer

이에 따라 TVlog.html 파일을 통해 얻을 수 있는 유의미하게 수사에 사용될 수 있는 아티팩트는 〔표 6〕과 같다.

〈Table 6〉 Artifact from TVLog.html

Tag Name	contents
CPersistentParticipantManager	원격 제어자에 해당하는 계정 정보(계정 닉네임, 계정 ID)
ModuleFileTransfer	원격 제어자가 제어 받는 디바이스로 전송한 파일 정보
triggerRSInfo	원격 제어자가 제어 받는 사용자에게 스크린샷 권한 요청 정보

4.2. AnyDesk

〈Table 7〉 Android AnyDesk Controlling device and Controlled device information

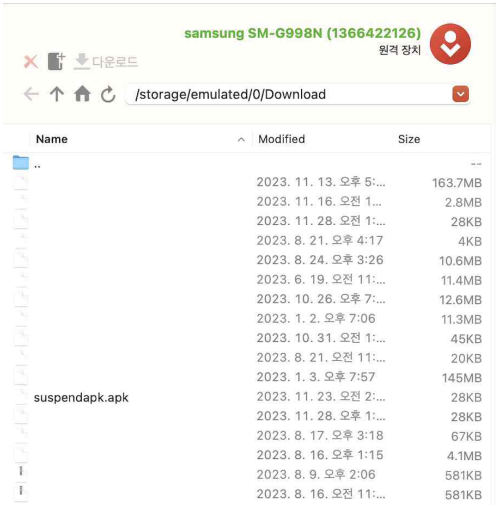
	AnyDesk ID	Version
Controlling Device	1 302 250 969	MacOS 7.3.0 (444a4c2f)
Controlled Device	1 366 422 126	Android 7.0.0 (5614704)

〔표 7〕에는 제어 받는 기기에서 원격 제어에 사용되는 AnyDesk ID와 설치된 AnyDesk의 버전 정보에 관한 내용이 담겨있다. AnyDesk의 파일 저장 위치는 /data/data/com.com.anydesk.anydeskandroid 이다. 해당 경로는 cache, code_cache, files, shared_prefs 폴더를 포함 하고 있는데 files 경로를 제외한 다른 모든 경로에는 아무런 파일이 저장되어 있지 않았다. Files 경로에 위치한 파일의 경로 정보에 대한 것은 〔표 8〕에서 확인할 수 있다.

〈Table 8〉 Android AnyDesk application data storage path and contents

Path	File Name	contents
data/com.com.anydesk.anydeskandroid/files	downloads/〔TimeStamp〕/〔downloaded File Name〕	원격 제어를 통해 원격 제어 받는 사용자의 기기에 다운로드 받아진 파일

AnyDesk도 마찬가지로 TeamViewer처럼 원격 제어가 이루어질 당시에는 제어자가 제어 받는 사용자의 디바이스에 자유롭게 파일 전송을 할 수 있다. 〔Figure 8〕은 제어자의 디바이스에서 AnyDesk를 통해 원격 제어를 진행하던 중 파일 전송 창을 띄운 모습이다.



〈Figure 8〉 AnyDesk File Transfer Service

만일, AnyDesk를 이용해 원격 제어가 범죄에 활용되어 제어자가 제어 받는 사용자의 디바이스에 악성 어플리케이션 파일을 전송하였을 것이다. 사용자에게 이를 설치하도록 유도하여 추가 범죄에 이 악성 파일을 활용했다고 했을 때 해당 경로에서 전송된 파일을 직접 확인하여 어떠한 악성 행위를 할 수 있는 파일인 지 추가적인 분석에 사용될만 한 증거로써 활약이 가능하다. 이 외의 경로에는 아무런 파일이 존재하지 않아서 추가적으로 아티팩트로 사용할 만한 정보는 없었다.

4.3. AirDroid

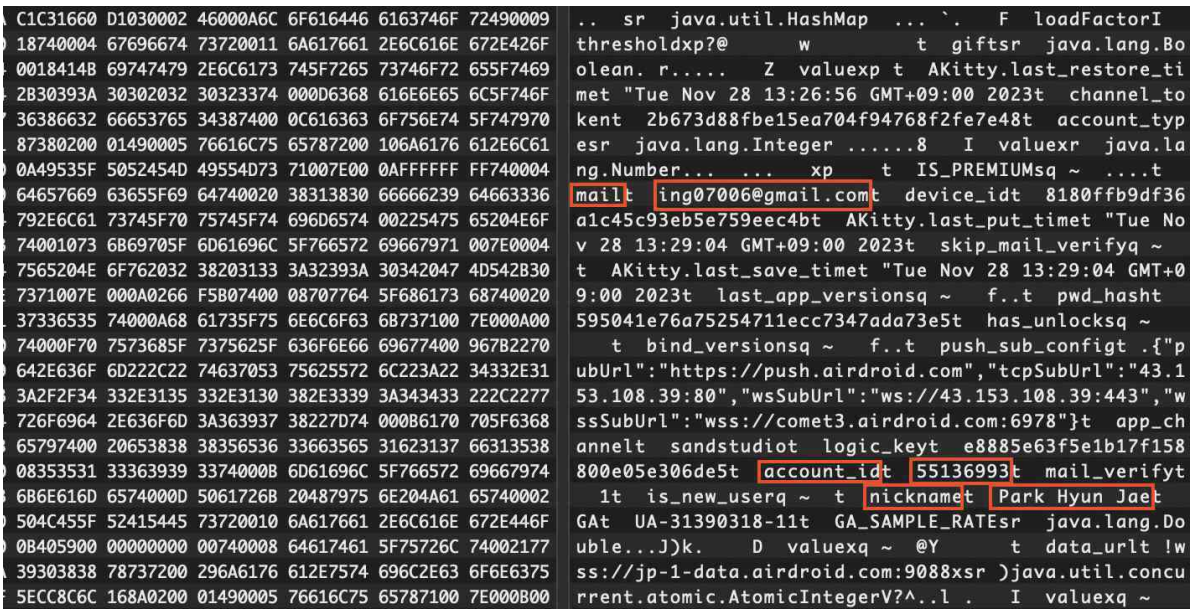
AirDroid는 앞선 두 개의 원격제어 앱과는 다른 방식으로 원격 제어를 제어한다. 바로 계정 단위로 원격 제어를 지원한다는 것이다. 별도의 원격 제어용 ID를 사용하지 않고 해당 서비스를 이용하기 위해 가입한 계정으로 제어할 기기와 제어 받을 기기에서 동일하게 로그인하게 되면 원격 제어가 가능한 형태로 운영하고 있는 것을 확인하였다. 따라서 아티팩트 조사를 위해 로그인 한 계정은 "ing07006@gmail.com" 의 이메일을 사용하여 로그인하였다. AirDroid 역시 /data/data/com.sand.airdroid 경로에 앱의 데이터 파일 등이 저장되어 있었다. 해당 경로 내의 파일들 중에서 유의미한 정보를 찾아보았다. 〔표 9〕은 해당 경로에서 찾은 유의미한 파일들과 파일이 의미하는 부분을 기록해둔 표이다.

〈Table 9〉 Android AirDroid application data storage path and contents

Path	File Name	contents
data/com.sand.airdroid/files	account_backup	원격 제어 서비스를 이용하기 위해 로그인한 계정의 닉네임과 아이디 정보
	account_preference	원격 제어 서비스를 이용하기 위해 로그인한 계정의 상세 정보
	main_preference_bk	원격 제어 서비스를 이용하기 위해 설정한 제어 받는 사용자의 디바이스에게 요청한 권한 및 계정 정보, 서비스 이용을 위한 세부 사항 정보 등
	recursive_file_index_phone	원격 제어 서비스에서 제어받는 사용자의 디바이스 파일 시스템 내부 파일 인덱스 리스트

data/com.sand.airdroid/databases	app.db	AirDroid 어플리케이션에서 사용하는 기기 내 설치된 패키지명, http post, get 요청에 대한 데이터베이스
	transfer.db	AirDroid를 통해 원격 제어가 이루어지는 당시 제어자에 의해 제어받는 디바이스로 전송되는 파일 목록에 대한 데이터베이스
data/com.sand.airdroid/shared_prefs	com.sand.airdroid_preference.xml	AirDroid 원격 제어 서비스를 이용하는 계정에 대한 상세 정보

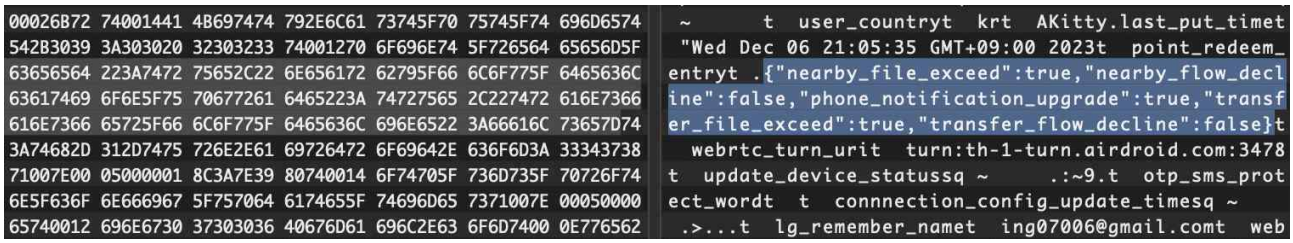
첫 번째로 data/com.sand.airdroid/files에 위치한 account_backup 파일에서 원격 제어 서비스를 이용하기 위해 로그인 한 계정의 닉네임 정보와 아이디 정보를 얻을 수 있을 수 있었다.



〈Figure 9〉 account_backup file contents from Hex Fiend

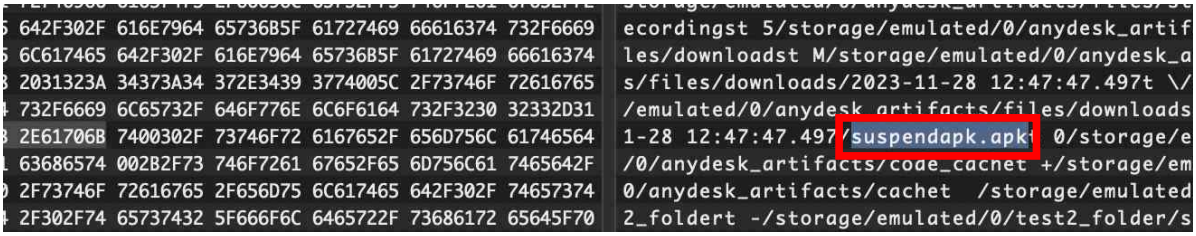
〔Figure 9〕은 해당 파일을 Hex Fiend에서 확인 하여 16진수 값에 해당하는 ASCII 문자열을 확인한 모습이다. 좌측이 불필요한 16진수 값은 일부 생략하여 표기하였다. 이 때 java.util.HashMap을 통해 일부 암호화 되었음을 알 수 있지만, 일부 변수와 변수에 해당하는 값은 그대로 보존되어 일부 아티팩트로 활용할 만한 정보를 얻어낼 수 있었다. 먼저 mail 이라는 변수에는 "ing07006@gmail.com" 의 로그인 계정의 이메일 정보를 온전히 확인할 수 있다. 그리고 account_id 변수가 보이는데 이는 "55136993" 으로 확인 가능하다. account_id 의 경우 원격 제어 서비스 자체에서 사용되지는 않지만 앱 내에서 자체적으로 사용하는 것으로 추정하고 있다. 이후 nickname 변수로 "Park Hyun Jae"가 확인된다. 이는 AirDroid 서비스를 회원가입 할 때 입력해야 하는 nickname을 출력 하기 위해 사용되는 변수이다. AirDroid 원격 제어 서비스에서 로컬의 계정 정보가 중요한 이유는 AirDroid는 동일한 계정으로 제어자와 제어를 받는 디바이스에서 로그인하여서 원격 제어를 하는 서비스이기 때문에 로컬에 남아 있는 계정 정보로 만일 원격 제어를 시도한 사람이 악성 행위를 저지른 상황 등에서 아티팩트로서 활용이 가능하기 때문이다. 해당 정보는 같은 경로 내의 account_preference_bk 파일에서도 동일하게 확인이 가능하다.

두 번째로 같은 경로 내에 위치한 main_preference_bk 파일이다. 해당 파일 역시 확장자가 존재하지 않아 바이너리를 열람할 수 있는 소프트웨어를 통해 파일 내부의 내용을 관찰하였다. 〔Figure 10〕를 통해 확인 가능한 main_preference_bk 파일에는 원격 제어 서비스를 이용하는데 필요한 세부 권한에 대한 정보이다. 마찬가지로 좌측에 불필요한 16진수 값은 일부 생략하였다. "nearby_file_exceed"에 해당하는 value 값은 true인데 이는 근처 기기에 대한 액세스 권한이 허용 되어 있음을 알 수 있다.



〈Figure 10〉 main_preference_bk file contents from Hex Fiend

"phone_notification_upgrade"에 해당하는 value 값은 true인데 이는 제어자의 디바이스에서 제어 받는 디바이스에서 오는 앱 알림을 수신하고 답장할 수 있는 권한을 부여하도록 하는 것이 현재 허용 되어 있다는 사실을 나타낸다. "transfer_file_exceed"의 경우 제어자의 디바이스에서 자유롭게 제어 받는 디바이스로 파일 전송을 허용하도록 하는 권한인데 이 역시도 현재 true로 지정되어 있다. 현재 원격 제어를 통해 제어 받는 사용자의 디바이스에 어떠한 권한이 허용 되어 있는 지를 파악하는 것은 매우 중요하다. 권한의 허용 정도에 따라 범죄 상황에 원격 제어 서비스가 활용 되었을 때 어느 수준의 악성 행위를 저지를 수 있고, 피해자의 피해 수준을 가늠할 수 있는 척도로 수사에서 활용할 수 있기 때문이다.



〈Figure 11〉 recursive file contents from Hex Fiend

다음으로 같은 경로 내의 recursive_file_index_phone 파일 내부를 확인하였다. 이는 〔Figure 11〕에서 마찬가지로 바이너리의 일부를 확인한 사진을 첨부하였다. 해당 파일에서는 제어 받는 디바이스의 파일 시스템에 저장된 파일들의 시퀀스가 저장되어 있었는데, 이 때 원격 제어 서비스를 통해 전송받은 파일 "suspendapk.apk" 도 확인할 수 있었다. 정확히 해당 파일 목록이 어떤 기준에 의해 나열 되었는지 확인할 수는 없지만 해당 파일의 타임스탬프가 함께 출력되고 있기 때문에 해당 파일이 원격 제어 당시에 전송 받은 파일인지에 대한 정보를 얻는 데에는 유용하게 사용될 수 있는 아티팩트이다.

그리고 data/com.sand.airdroid/databases 경로에 저장된 확장자가 db인 파일을 열람하였다. 확장자가 db인 파일은 데이터베이스 포맷 파일로 데이터베이스 형식 파일을 열람 가능한 소프트웨어가 필요하다. 이번 실험에서는 SQLite Viewer Web App available for web로 웹 상에서 간편하게 데이터베이스 포맷 파일을 열람할 수 있는 온라인 웹 사이트에서 해당 파일들을 열람하였다. 먼저 app.db 파일의 경우 테이블의 개수는 총 27개로 확인하였다. 테이블의 주요 내용으로는 AirDroid 어플리케이션에서 수집하는 디바이스 내에 설치된 모든 패키지 정보에 대한 테이블, 원격 제어가 이루어지는 동안 발생했던 작업의 이름과 요청의 종류에 대한 테이블, Message를 SEND하거나 RECEIVE 한 경우에 대한 테이블 등이 있다. 해당 테이블 들을 직접 확인함으로써 원격 제어 당시에 어떤 종류의 작업이 발생하였고, 해당 작업에 대한 타임스탬프 등의 기록을 보다 더 상세히 확인 가능하다. 먼저 AirDroid 어플리케이션에서 수집하는 디바이스 내에 설치된 모든 패키지 정보에 대한 테이블은 〔Figure 12〕에서 확인이 가능하다. "PACKAGE_ID" 컬럼에서는 해당 패키지의 풀 네임을 확인할 수 있고 "NAME" 컬럼에서는 해당 패키지가 어떤 이름의 어플리케이션으로 설치되어 있는 지를 기록하는 컬럼이다. 〔Figure 13〕에는 원격 제어가 이루어질 동안 어떤 작업이 수행 되었고, 해당 작업이 요청한 명령의 종류에 대한 테이블인 FLOW_STAT TABLE에 해당하는 사진이다. "ACTION_NAME" 컬럼을 보면 원격 제어 서비스 내에서 지정한 이름에 해당하는 서비스들을 확인할 수 있는데, "/msg/get" 등의 액션명은 메시지를 GET으로 요청한 등의 작업임을 유추할 수 있다.

app.db ▶ APP_CACHE

Reset Filters | Records: 512

	_id	PACKAGE_ID	NAME	VERSION_CO...
	Search column...	Search column...	Search column...	Search column...
1	1	com.google.android.networkstack.tethering	Tethering	34
2	2	com.samsung.android.provider.filterprovider	Filter Provider	522100000
3	3	com.sec.android.app.DataCreate	Automation Test	2
4	4	com.android.cts.priv.ctsshim	com.android.cts.pr...	31
5	5	com.samsung.android.smartswitchassistant	com.samsung.and...	211800000
6	6	com.sec.vsim.ericssonnsds.webapp	NSDSWebApp	200600000
7	7	com.sec.android.app.setupwizardlegalprovi...	SetupWizardLegal...	210400000
8	9	com.samsung.android.app.galaxyfinder	파인더	994100000
9	10	com.sec.location.nsflp2	Samsung Location...	604900000
10	12	com.samsung.android.themestore	Galaxy Themes	520504103
11	13	com.sec.android.app.chromecustomizations	ChromeCustomiza...	302900000

〈Figure 12〉 APP_CACHE TABLE of app.db file from SQL Viewer web applicaion

이러한 작업들은 CREATED_TIME 컬럼에서 앱 내에서 자체 환산한 타임 스탬프로 시간 순대로 정렬이 가능해서 작업의 진행 흐름을 볼 수 있는 증거물로써 활용이 가능하다.

app.db ▶ FLOW_STAT

Reset Filters | Records: 72 | Search 72 records...

	_id	ACTION_NAME	LENGTH	FORMAT_SIZE	MODULE	CREATED_TIME
	Search column...	Search column...	Search column...	Search column...	Search column...	Search column...
1	2	flow_total	18635828	17.77 MB	total	1701145616970
2	93	/p20/config/get	17936	18 KB	GET	1701853956180
3	94	/phone/dataflow	24832	24 KB	POST	1701853956215
4	95	/p20/config/get	12768	12 KB	GET	1701853957156
5	96	/msg/get	461	461 byte	GET	1701853957297
6	97	push_receive	4622	5 KB	PUSH	1701853957305
7	98	/phone/dataflow	64	64 byte	POST	1701853957444
8	99	/msg/get	4026	4 KB	GET	1701855697632
9	100	/p20/device/updat...	5040	5 KB	GET	1701863666352
10	101	/p20/device/updat...	288	288 byte	GET	1701863667252
11	102	/	396	396 byte	GET	1701863667343
12	103	/push	1811	2 KB	POST	1701863671796
13	104		67	67 byte	GET	1701863671832
14	105	/phone/connect	861	861 byte	POST	1701863672179
15	106	/	249	249 byte	GET	1701863672183
16	107		26	26 byte	GET	1701863672580
17	108	forward_data	36625	36 KB	FORWARD	1701863672661
18	109	forward_data	49501	48 KB	FORWARD	1701863672669
19	110	/forward/data	360	360 byte	POST	1701863672757

〈Figure 13〉 FLOW_STAT TABLE of app.db file from SQL Viewer web applicaion

〔Figure 14〕에는 MSG_SEND_RECORD를 PUSH한 것에 대한 데이터들이 존재하는 PUSH_MSG_SEND_RECORD TABLE의 모습이다. 이 때 해당 테이블에서는 ip에 해당하는 정보와 port에 해당하는 정보가 존재하는데. 해당 ip와 port 정보는 CONTENT 필드의 데이터 중에서 "deviceid"의 value 값을 통해 제어 받는 기기에 해당하는 정보임을 알 수 있었다.

다음으로 같은 경로 내의 transfer.db 파일을 열람하였다. 해당 데이터베이스 파일 내에는 3개의 테이블이 존재한다. 이 테이블 들 중 유의미한 아티팩트로 사용될 만한 테이블은 transfers 테이블이다. 해당 테이블에는 어떤 채널을 사용하여 어느 경로로 제어자가 제어 받는 사용자의 디바이스에 파일을 전송하였는 지가 기록되어 있다.

app.db ▶ PUSH_MSG_SEND_RECORD

Reset Filters | Records: 16 | Search 16 records...

	_id	CONTENT	CREATE_TIME
1	1	{"getkey":""}	1701145745266
2	2	{"pid":1701145745242,"uri":"/bind_state/changed/"} ==== RESULT : === push Msg return : {"ret":0,"subCo...	1701145747543
3	3	{"getkey":""}	1701145825215
4	4	{"forward_ok":false,"http_ok":true,"netOpts":{"ip":"172.21.81.249","port":8888,"socket_port":8889,"ssl_p...	1701145829866
5	5	{"getkey":""}	1701145830127
6	6	{"pid":1701145824,"uri":"/cfunc/server_info_response/","result":"{\"model\":\"SM-G998N\",\"manu\":\"sam...	1701145830539
7	7	{"getkey":""}	1701146095483
8	8	{"pid":1701146095,"uri":"/cfunc/server_info_response/","result":"{\"model\":\"SM-G998N\",\"manu\":\"sam...	1701146098484
9	9	{"getkey":""}	1701863671784
10	10	{"pid":1701863667,"uri":"/cfunc/server_info_response/","result":"{\"model\":\"SM-G998N\",\"manu\":\"sam...	1701863673583
11	11	{"getkey":""}	1701863674423
12	12	{"pid":1701863672,"uri":"/cfunc/server_info_response/","result":"{\"model\":\"SM-G998N\",\"manu\":\"sam...	1701863676486
13	13	{"getkey":""}	1701863676575
14	14	{"pid":1701863673,"uri":"/cfunc/server_info_response/","result":"{\"model\":\"SM-G998N\",\"manu\":\"sam...	1701863678177
15	15	{"getkey":""}	1701863678256
16	16	{"forward_ok":true,"http_ok":true,"imsi":"","netOpts":{"ip":"172.21.81.231","port":8888,"socket_port":888...	1701863679389

〈Figure 14〉 PUSH_MSG_SEND_RECORD TABLE of app.db file from SQL Viewer web applicaion

이 역시도 앞선 데이터베이스의 테이블에서 확인했던 deviceid를 통해 제어 받는 사용자의 디바이스임을 인지할 수 있었다. 해당 원격 제어 중에 전송된 파일에 해당 하는 정보 역시 transfers 테이블에 상세히 저장되어 있는데 파일의 크기와 파일명, 파일이 생성된 시각, 파일의 unique_id, transfer 작업의 pid 등의 파일에 해당하는 정보 역시도 상세히 저장되어 있다. 다만 파일이 생성된 시각 역시 앱 내에서 자체 변환된 시각으로 기록되어 있다. transfer 작업의 process id에 해당하는 정보는 앞선 테이블에서 key에 해당하는 값 중에서 "pid"가 있었는데 이 key 값에 동일한 값이 존재한다면 file transfer 와 연관 지어 제어자가 파일 전송한 작업에 대한 추가적인 정보로 연결 지을 수도 있다. 해당 테이블에 대한 정보는 〔Figure 15〕에서 자세히 확인할 수 있다. 이 때 칼럼에 대한 정보는 첨부 이미지의 크기가 한정되어 있어 일부 생략 되었다.

transfer.db ▶ transfers

4 | Reset Filters | Records: 4 | Search 4 record

	device_id	path	title	created_time	transfer_pid	target_name
1	8180ffb9df36a1c4...	/sdcard/Download/...	suspendapk2.apk	1701146089033	1701146089029	Web
2	8180ffb9df36a1c4...	/sdcard/airdroid/u...	python.zip	1701863711210	1701863711208	Web
3	8180ffb9df36a1c4...	/sdcard/airdroid/u...	아차차 전자출결+ 1.0...	1701863746545	1701863746545	Web
4	8180ffb9df36a1c4...	/storage/emulated/...	suspendapk2.apk	1701146562372	1701146562371	Web

〈Figure 15〉 transfers TABLE of transfer.db file from SQL Viewer web applicaion

다음으로 data/com.sand.airdroid/shared_prefs 경로 내의 com.sand.airdroid_preference.xml 파일을 열람하였다. 해당 파일에는 AirDroid 서비스를 이용하는 계정의 상세 정보가 담겨있다. 예를 들어 AirDroid 서비스에 로그인 되어 있는 계정에서 AirDroid에서 제공하는 제어자의 디바이스를 통한 제어 받는 기기의 원격 파일 관리 서비스를 사용하는 지에 대한 허용 여부, 카메라를 통해 제어자가 제어 받는 사용자의 장치의 카메라로 장치의 주변을 확인할 수 있는 지에 대한 허용 여부, 또 제어자가 제어 받는 사용자의 디바이스의 스크린을 원격으로 미러링한 화면을 열람할 수 있는 지에 대한 허용 여부, 제어자의 기기에서 제어 받는 디바이스에 오는 알람을 동기화 시켜 대신 알람을 들을 수 있는 지에 대한 허용 여부, 원격으로 디바이스의 연락처를 관리할 수 있는 것에 대한 허용 여부, 웹에서 제어 받는 장치의 상세한 위치를 원격으로 찾을 수 있는

지에 대한 허용 여부 등이 담겨있다. 특히 카메라와 스크린 미러링 마이크의 사용 허가를 통해 장치 주변의 소리를 듣고 제어자가 장치 주변에 있는 것처럼 활용 가능한 오디오 권한에 대한 허용 여부 역시 별도로 확인할 수 있다. 해당 허용 여부를 통해 제어 받는 사용자가 디바이스에서 직접적으로 출력되는 화면 뒤인 백그라운드에서 제어자가 어떤 종류의 작업을 수행할 수 있었는 지에 대한 정보를 상세히 얻을 수 있다. 만일 원격 제어가 범죄에 사용되었다면 범인이 해당 디바이스를 통해 백그라운드 상에서 앞서 언급했던 권한 허용되어 있다면 해당 정보에 관해서 디바이스에 저장된 어떤 정보까지 범인에게 넘어갔음을 판단하는 척도로 활용할 수 있다. 특히나 마이크를 사용하는 권한 여부도 알 수 있기 때문에 장치가 도청 행위가 이루어졌을 수도 있음을 판단하는 아티팩트로서 활용 가능하다. 해당 파일의 내용은 〔Figure 16〕를 통해 자세히 확인할 수 있다.

com.sand.airdroid_preferences.xml

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="otp_sms_disable_country_list">ru</string>
    <boolean name="otp_sms_free_user_enable" value="false" />
    <long name="contact_last_modify_time" value="1701149482639" />
    <string name="acra.user.email">ing07006@gmail.com</string>
    <boolean name="remote_contact_enable" value="true" />
    <boolean name="otp_is_user_premium" value="false" />
    <int name="cache_state" value="3" />
    <string
        name="remote_permission_55136993">
        {&amp;quot;airmirror&amp;quot;:true,&amp;quot;camera&amp;quot;:true,&amp;quot;camera_audio&amp;quot;:true
        ,&amp;quot;contacts&amp;quot;:true,&amp;quot;file&amp;quot;:true,&amp;quot;findphone&amp;quot;:true
        ,&amp;quot;notification&amp;quot;:false,&amp;quot;screen&amp;quot;:true,&amp;quot;screen_audio&amp;quot;:true
        ,&amp;quot;sms&amp;quot;:false,&amp;quot;code&amp;quot;:0}</string>
    <boolean name="remote_sms_enable" value="false" />
    <string name="pref_c2c_des_key">8180e888</string>
    <string name="otp_sms_protect_word"></string>
    <boolean name="remote_camera_enable" value="true" />
    <int name="ssl_certificate_vertify" value="-1" />
    <string name="pref_des_key">66bba18095a34836</string>
    <string name="otp_sms_protect_help">SMS는 '인증 코드 보호'로 숨겨집니다. 자세히 알아보기:
        www.airdroid.com/verification-code.html</string>
    <string name="account_id">55136993</string>
    <string
        name="fcm_registration_id">cgIBQ06BWIU:APA91bFlVMw6sDelJCVc3LXp8A-fCkAFWm4Sf5kyvHaqjMqkiqzlzzzYmR-P_HG1k
        EeWwJsTahJEWfS4OkqOSZMzV84xdvjOYXxnKH2v--c7qVGUgWJ_4cadRM9i5FMw9hd0IJtiSP3B</string>
    <long name="otp_sms_protect_word_time" value="0" />
    <int name="ContactState" value="1" />
    <boolean name="remote_file_enable" value="true" />
</map>
```

〈Figure 16〉 com.sand.airdroid_preferences.xml file content

Ⅴ. 주요 아티팩트 정리 및 수사 활용 방안

5.1. 주요 아티팩트 정리

4장에서 국내 보이스피싱 범죄에서 주로 활용되는 원격 제어 어플리케이션 TeamViewer, AnyDesk, AirDroid 3종에서 애플리케이션 데이터 파일에서 얻을 수 있는 아티팩트의 분석을 진행했다. 세 가지 어플리케이션에서 일관되게 도출할 수 있는 주요 정보 요소는 원격 제어자가 원격 제어를 받는 사용자의 기기에 전송한 파일의 경로와 그 파일 자체이다. 또한 애플리케이션 데이터 경로에 별도 데이터를 저장하지 않는 AnyDesk를 제외한 TeamViewer와 AirDroid에서는 원격 제어자를 특정할 수 있는 ID 및 로그인 계정 정보, 닉네임 정보 등에 대한 정보를 통해 원격 제어가 끝난 뒤에도 원격 제어자를 특정지울 수 있다. 또한 이 두 개의 어플리케이션에서는 원격 제어를 진행할 당시에 해당 앱에서 요청한 권한에 대한 허용 여부를 알 수 있는 정보가 저장되어 원격 제어 당시에 제어자가 디바이스에 저장된 민감 정보 등을 어떤 수준까지 파악이 가능한 지를 가늠할 수 있는 정보 역시 얻을 수 있었다. 3종의 어플리케이션 중에서 AirDroid를 통해 얻어낼 수 있는 아티팩트가 가장 많았는데 이는 다른 2종의 원격 서비스와는 달리 자신의 계정으로 직접 자신의 기기를 제어하는 방식으로 진행하는 것이기 때문으로 추측된다. 따라서 AirDroid 서비스는 제어자, 피제어자 2개의 계정으로 이루어진 것이 아닌 자신이 직접 자신의 계정으로 제어 서비스를 활용하는 것으로 간주되어 해당 디바이스에 더 많은 정보를 저장하는 것으로 추정하고 있다.

하지만 원격 제어의 계정 정보 외에 제어자의 IP 정보 같이 더 세밀하게 특정 지을 수 있는 정보를 얻을 수

없었다. TeamViewer, AnyDesk, AirDroid 모두 해당 원격 제어 서비스를 통해 원격 통신이 이루어지므로 각 서비스의 각각의 라우터 및 프록시 서비스를 통과함에 따라 로그를 통해 확인한 IP 주소에서는 원격 제어를 받는 기기에 해당하는 IP 주소만 얻어내었고 원격 제어자의 IP 주소는 얻어내기 어려웠다.

각 어플리케이션의 주요 아티팩트 요약은 다음 〔표 10〕과 같다. 이후 원격 제어 어플리케이션 설치를 유도한 실제 보이스피싱 사례를 기반으로 시나리오를 가정하고, 해당 상황에서 수집된 아티팩트가 수사에 활용될 수 있는 방안을 제시하겠다.

〈Table 10〉 Each used Remote control application important data storage path and contents

Application	Path and File Name	contents
TeamViewer	data/com.teamviewer.quicksupport.market/sharedprefs/com.teamviewer.quicksupport.market_preferences.xml	원격 제어 당시의 QuickSupport 어플리케이션에서 요청한 권한 등
	data/com.teamviewer.quicksupport.market/files/client.conf	유저의 SUID 정보
	data/com.teamviewer.quicksupport.market/files/TVLog.html	원격 제어의 시작 단계부터 종료까지 TeamViewer 작업 이벤트 로그
AirDroid	data/com.sand.airdroid/files/account_backup	원격 제어 서비스를 이용하기 위해 로그인한 계정의 닉네임과 아이디 정보
	data/com.sand.airdroid/files/account_preference	원격 제어 서비스를 이용하기 위해 로그인한 계정의 상세 정보
	data/com.sand.airdroid/files/main_preference_bk	원격 제어 서비스를 이용하기 위해 설정한 제어 받는 사용자의 디바이스에게 요청한 권한 및 계정 정보, 서비스 이용을 위한 세부 사항 정보 등
	data/com.sand.airdroid/files/recursive_file_index_phone	원격 제어 서비스에서 제어받는 사용자의 디바이스 파일 시스템 내부 파일 인덱스 리스트
	data/com.sand.airdroid/databases/app.db	AirDroid 어플리케이션에서 사용하는 기기 내 설치된 패키지명, http post, get 요청에 대한 데이터베이스
	data/com.sand.airdroid/databases/transfer.db	AirDroid를 통해 원격 제어가 이루어지는 당시 제어자에 의해 제어받는 디바이스로 전송되는 파일 목록에 대한 데이터베이스
	data/com.sand.airdroid/shared_prefs/com.sand.airdroid_preference.xml	AirDroid 원격 제어 서비스를 이용하는 계정에 대한 상세 정보
AnyDesk	data/com.com.anydesk.anydeskandroid/files/downloads/〔TimeStamp〕/〔downloaded File Name〕	원격 제어를 통해 원격 제어 받는 사용자의 기기에 다운로드 받아진 파일

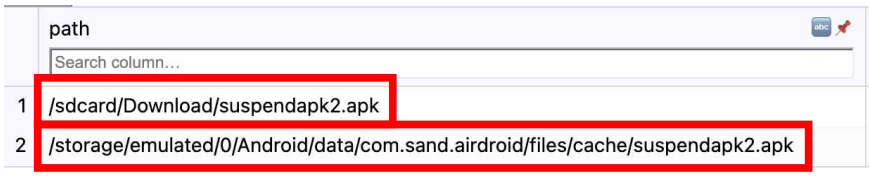
5.2. 수사 활용 방안

원격 제어가 범죄에 활용된 경우 원격 제어자는 범인이고 원격 제어를 받는 사용자는 피해자가 된다. 이 때 수사가 진행 되는 시점은 피해자가 이 원격 제어 행위가 현재 보이스피싱 행위임을 인지하지 못하는 상황이므로 원격 제어가 종료된 상황이다. 앞서 수집했던 원격 제어 어플리케이션의 아티팩트를 활용하여 수사에 활용할 수 있다. 원격 제어 어플리케이션 중 TeamViewer를 이용한 피싱 사건〔4〕를 예로 들면, 범인은 피해자의 휴대폰을 원격 제어하며 피해자에게는 이를 인지하지 못하게 하는 방식으로 범행을 저질렀다. 이와 같은 사건을 경찰이 수사할 때 연구를 통해 수집한 아티팩트를 통해 범인을 특정 지을 수 있는 TeamViewer 로그인

계정 정보를 얻어낼 수 있다. 해당 계정 정보 등을 통해 범인 혹은 용의자를 매우 빠르게 특정 지을 수 있다. 또한 닉네임이 만일 범인의 실명으로 되어 있다면 수사에 한층 속도를 낼 수 있도록 큰 도움을 줄 수 있다. 그리고 범인은 위의 원격제어 사례에서도 그렇듯 피해자에게 원격 제어 받는 기기에서 관심을 끄도록 유도했기 때문에 정확히 어떠한 작업이 이루어졌는 지를 알 수 없다. 그리고 기기 내에 저장된 민감 정보가 어느 정도의 수준까지 범인에게 유출 되었는지도 알 수 없다. 이 때 피해자의 기기의 정보를 백업해두고 루트 권한을 획득한 뒤에 TeamViewer 앱의 데이터 경로에 위치한 아티팩트를 수집하여 TVLog.html 파일을 얻어낸다. 해당 파일의 분석을 통해 원격 제어 작업의 흐름을 파악하고, preference 파일을 통해 어떤 권한이 허용 되었는 지를 확인하여 추가 피해를 미리 예방 하는 등에 도움을 줄 수 있다. 다음은 원격 제어 어플리케이션들에서 획득한 다양한 아티팩트를 활용한 가상의 시나리오를 작성하였다.

5.2.1. 시나리오 1

시나리오 1은 원격 제어 어플리케이션 중 AirDroid를 사용하여 보이스피싱 피의자가 피해자의 휴대폰에 자신의 계정으로 로그인하도록 유도하고, 원격 연결을 시도하여 추가 악의적인 행위를 할 수 있는 "suspendapk2.apk" 파일을 전송 후 설치까지 진행한 것이다. 안드로이드 디바이스에서 이 시나리오를 수행한 결과 원격으로 전송 받은 파일 목록에 대한 db 파일을 획득할 수 있다. 〔Figure 17〕은 AirDroid에서 수집된 아티팩트 중 원격으로 전송받은 파일 목록에 관한 데이터베이스와 관련된 그림이다. 〔Figure 18〕은 보이스피싱 피의자의 계정 정보가 담긴 아티팩트와 관련된 그림이다. 해당 아티팩트들을 통해서 피의자가 AirDroid에 어떤 계정을 사용했는 지에 대한 계정 정보와 피해자 휴대폰에 어떤 악성 파일들을 전송했는 지에 대한 정보를 수집할 수 있다. 전송되어 설치된 악성 앱이 피해자의 휴대폰에 설치된 채로 남아있다면 해당 앱을 추가로 분석하여 피해자의 디바이스에 어떠한 악영향을 미쳤는지에 대한 추가 분석으로 진행될 수 있다.



	path
	Search column...
1	/sdcard/Download/suspendapk2.apk
2	/storage/emulated/0/Android/data/com.sand.airdroid/files/cache/suspendapk2.apk

〈Figure 17〉 data/com.sand.airdroid/databases/transfer.db file content

```
<string name="otp_sms_disable_country_list">ru</string>
<boolean name="otp_sms_free_user_enable" value="false" />
<string name="acra.user.email">ing07006@gmail.com</string>
<boolean name="remote_contact_enable" value="true" />
<boolean name="otp_is_user_premium" value="false" />
<int name="cache_state" value="3" />
```

〈Figure 18〉 data/com.sand.airdroid/shared_prefs/com.sand.airdroid_preference.xml

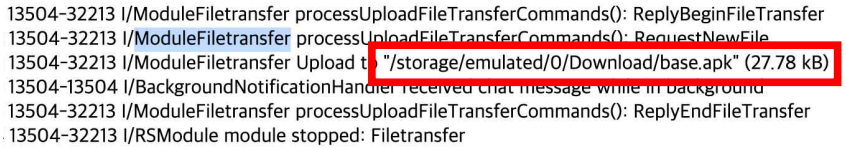
5.2.2. 시나리오 2

시나리오 2는 원격 제어 어플리케이션 중 TeamViewer를 사용하여 보이스피싱 피의자가 피해자에게 자신의 접속 코드를 입력하도록 하여 원격 연결을 시작하고, 원격 연결이 시작된 뒤 추가 악의적인 행위를 할 수 있는 "base.apk" 파일을 전송하는 것이다. 안드로이드에서 이 시나리오를 실행한 결과 피의자의 TeamViewer ID 및 계정 이름, 원격 제어가 이루어지는 동안 작업 로그 파일을 획득할 수 있다. 〔Figure 19〕와 〔Figure 20〕은 TeamViewer에서 수집된 아티팩트 중 피의자의 계정 정보와 작업 로그 등을 확인할 수 있는 아티팩트와 관련된 그림이다. 해당 아티팩트를 통해 피의자 원격 제어를 위해 사용한 계정의 고유 ID 정보, 계정이 이름 정보를 얻을 수 있다. 또한, 모듈 단위로 작업 로그가 기록되므로 파일 전송 모듈이 실행된 시점의 로그를 살펴보면 원격으로 피해자의 휴대폰에 전송된 파일의 경로와 파일에 대한 정보를 얻을 수 있다. AirDroid의 경우와 마찬가지로 전송된 파일이 추가적인 악성 행위를 행할 수 있는 지에 대한 조사로 이어질

수 있다.

〈Figure 19〉 TeamViewer ID and name of data/com.teamviewer.quicksupport.market/files/TVLog.html

〈Figure 20〉 Filetransfer Module content of TVLog.html



VI. 결 론

신종 코로나바이러스 감염증(코로나19)로 인한 비대면 환경은 보이스피싱의 패러다임을 변화시켰다. 전통적인 전화 기반의 보이스피싱에서 벗어나, 현재는 메신저 및 원격 제어 서비스를 통한 피싱 기술이 발전하고 있다[12]. 특히, TeamViewer와 같은 원격 접속 앱을 악용하여 피해자의 모바일 기기를 조종하여 피해자의 명의로 가칭된 비대면 계좌를 개설하는 등의 새로운 피싱 형태가 등장했다[13]. 이와 함께 원격 제어 어플리케이션은 어디에서나 설치된 앱을 통해 태블릿이나 스마트폰을 통한 원격 PC 접속을 용이하게 해주는 앱으로, 이전에는 주로 직장인들 사이에서 각광받았다. 그러나 이러한 편리한 접속 기능이 피싱 사기에 이용되어, 피해자로 하여금 앱 설치를 유도하고 이를 통해 돈을 빼앗아가는 사례가 증가하고 있다. 범죄에 원격 제어 서비스가 사용되면 원격 제어 서비스 운영자는 사전에 경고한 문구 등을 바탕으로 이를 책임지려고 하지 않을 수 있기 때문에 본인이 모든 것을 증명해야하는 경우 역시 충분히 발생할 수 있다. 따라서, 이를 입증하기 위한 데이터를 획득하려면 원격 제어 애플리케이션이 디바이스에 저장하는 데이터의 내용을 이해하고 분석하는 것이 필수적이다.

본 연구에서는 전 세계에서 가장 널리 사용되는 원격 제어 어플리케이션인 TeamViewer, AnyDesk, AirDroid를 대상으로 분석을 수행하였다. 그리고 각 어플리케이션에 대해 아티팩트를 수집하고 분석을 수행하였다. 이를 통해 디바이스 내에 저장되는 데이터가 어떤 것이고 그것이 어떤 경로로 저장되는지 파악할 수 있었다. 이를 바탕으로 주요 아티팩트를 정리하였다. 분석된 데이터를 통해 원격 제어를 하는 제어자의 계정 정보 및 닉네임, 제어자가 디바이스로 전송한 파일 등을 얻을 수 있었다, 또한 어플리케이션에 따라 다르지만 원격 제어 당시에 어플리케이션에서 요청한 권한을 허용한 지에 대한 여부, 대략의 원격 제어 작업 로그 등을 확인할 수 있다. 본 연구에서는 각 어플리케이션들의 아티팩트 위치와 내용을 제시함으로써 보다 신속하고 효율적인 분석 및 증거 수집에 기여할 것으로 예상된다.

원격 제어자의 IP 정보나 네트워크 프로토콜에 대한 상세 정보를 얻을 수 있다면 원격 제어가 이루어지는 동안 범인이 어떤 행위를 하였는지 보다 명확히 파악할 수 있을 것 이다. 하지만 작업 로그에서도 네트워크 연결은 해당 원격 제어 서비스의 프록시 네트워크 및 많은 라우터 등을 거쳐 통신이 이루어지기 때문에 상세 네트워크 접속 정보 등을 확인할 수 없다. 따라서 향후 아티팩트 분석에서는 실시간으로 이루어지는 원격 제어 현장에 대한 분석을 통해 어떤 과정을 통해 네트워크 연결이 맺어지고, 맺어진 네트워크 연결 사이에서 얻을 수 있는 프로토콜 정보 등에 대한 연구가 필요하다.

참 고 문 헌 (References)

[1] Pashchenko, Denis. "fully remote software development due to COVID factor: Results of industry research (2020)." International Journal of Software Science and Computational Intelligence (IJSSCI), 13.3, pp.64-70, 2021.

[2] Drugarin, CV Anghel, Silviu Draghici, and Eugen Raduca. "Team Viewer Technology for Remote Control of a Computer." Analele Universitatii'Eftimie Murgu', 23.1, pp.61-66, 2016.

[3] Hyunmin Shin. "A Study on Improving Responses to Telecommunication Financial Fraud through Crime Script Analysis." Master's Thesis, Graduate School of Police Administration, Police University, 2022, Seoul.

[4] Kang-Jun Lee, "'Installed an App Sent by My Daughter and the Phone Started Acting on Its Own...' Lost 30 Million Won", moneytoday news, Available: https://news.mt.co.kr/mtview.php?no=2020113009340472659, 2024.01.11., confirmed.

[5] Eunbi Kim and Ikrae Jeong. "A Study on Improving User Verification for Non-face-to-face Financial Transactions - Focusing on Messenger Phishing Cases." Journal of the Korea Institute of Information Security and Cryptology, vol. 33, no. 2, pp. 353-362, 2023.

[6] Lahaie Colby, and David Leberfinger. "TeamViewer Forensics." pp.3, 2013.

[7] Manson, Jonathan. "Remote Desktop Software as a forensic resource." Journal of Cyber Security Technology, 6.1-2, pp.1-26, 2022.

[8] Magisk Android App. Available:https://github.com/topjohnwu/Magisk, 2024.01.11., confirmed.

[9] busybox Docker Image, Available:https://hub.docker.com/_/busybox, 2024.01.11., confirmed.

[10] Apktool PC, Available:https://apktool.org/, 2024.01.11., comfirmed.

[11] TWRP PC, Available:https://twrp.me/, 2024.01.11., confirmed.

[12] Young-ho Jeong and Hyeong-jun Ha. "A Study on the Current State and Countermeasures of Messenger Phishing Crimes." Journal of Criminal Investigations, vol. 8, no. 1, pp. 31-54, 2022, doi: 10.46225/CIS.2022.06.8.1.31.

[13] Jun-Woo Kwon, ""'Installed Just One App...' All Information in the Phone Went to the Phishing Criminal", yna news, Available:https://www.yna.co.kr/view/AKR20230525053000061, 2024.01.11. confirmed.

저 자 소 개

박 현 재 (Hyun-jae Park)
준회원

2019년 3월~현재 : 아주대학교 사이버보안학과 학사과정
관심분야 : 디지털 포렌식, 정보보호 등

손 태 식 (Taeshik Shon)
정회원

2000년 : 아주대학교 정보및컴퓨터공학부 졸업(학사)
2002년 : 아주대학교 정보통신전문대학원 졸업(석사)
2005년 : 고려대학교 정보보호대학원 졸업(박사)
2004년~2005년 : University of Minnesota 방문연구원
2005년~2011년 : 삼성전자 통신·DMC 연구소 책임연구원
2017년~2018년 : Illinois Institute of Technology 방문교수
2011년~현재 : 아주대학교 정보통신대학 사이버보안학과 교수
관심분야 : Digital Forensics, ICS/Automotive Security