*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.32, NO.1, Feb. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.1.11

# Electron App의 메시지 획득 방안에 관한 연구: 협업 툴 잔디, 슬랙, 팀즈 중심으로

김 성 수,[1†] 이 성 진[2‡]
[1,2]금융감독원(검사역, 선임검사역)

# A Study on Message Acquisition from Electron Apps: Focused on Collaboration Tools such as Jandi, Slack, and Microsoft Teams

Sung-soo Kim,[1†] Sung-jin Lee[2‡]
[1,2]Financial Supervisory Service (Examiner, Senior Examiner)

## 요 약

코로나19 이후 비대면 근무가 증가함에 따라 협업 툴의 사용이 증가하고 있다. 협업 툴은 다양한 기기에서의 접근성을 보장하기 위해 크로스 플랫폼(cross-platform) 형태로 개발되고 있으며, 이를 위해 Chromium 기반의 오픈소스 프레임워크인 Electron을 사용하는 것이 최신의 개발 경향이다. Electron 오픈소스 프레임워크를 사용하여 개발된 어플리케이션 Electron App의 경우 Chromium 기반 웹 브라우저와 유사한 방식으로 데이터를 저장하므로, 웹 브라우저에서 데이터를 획득하는 방법과 동일한 방법으로 어플리케이션의 데이터를 획득할 수 있다. 본 논문에서는 협업 툴 잔디(jandi), 슬랙(slack), 팀즈(microsoft teams)를 중심으로, 웹 브라우저 저장소(web storage)의 데이터 구조를 분석하고 이를 기반으로 Electron App의 메시지를 획득하는 방안을 제시한다. 잔디는 기존에 개발된 도구를 활용하여 Cache에서 메시지를 획득하였고, 슬랙, 팀즈의 경우 본 논문에서 개발한 메시지 카빙 도구를 이용하여 IndexedDB에서 메시지를 획득하였다.

## ABSTRACT

Collaboration tools are used widely as non-face-to-face work increases due to social distancing after COVID-19. The tools are being developed in a cross-platform manner with ‘Electron’, an open source framework based on Chromium, to ensure accessibility on multiple devices. Electron Apps, applications built with Electron framework, store data in a manner similar to Chromium-based web browsers, so the data can be acquired in the same way as the data is acquired from a web browser. In this paper we analyze the data structure of web storage and suggest a method to get the message from Electron Apps focused on collaboration tools such as Jandi, Slack, and Microsoft Teams. For Jandi, we get the message from Cache by using previously developed tools, and in the case of Slack and Microsoft Teams, we get the message from IndexedDB by using the message carving tool we developed.

**Keywords:** Electron App, Collaboration Tool, Jandi, Slack, Microsoft Teams

Received(09. 24. 2021), Modified(12. 06. 2021), Accepted(12. 26. 2021)
논문의 내용은 저자 개인의 의견이며 소속 기관의 공식 견해가 아님을 밝힙니다.

† 주저자, c0msherl0ck@gmail.com
‡ 교신저자, blackostrich@gmail.com(Corresponding author)

## I. 서 론

코로나19의 여파로 비대면 근무가 증가함에 따라 기업의 협업 툴 도입이 급격하게 증가하고 있다[1]. 이는 기업형 범죄 발생 시 협업 툴에서 주요 증거를 발견할 확률이 높다는 것을 의미한다. 특히, 협업 툴에서 송·수신한 메시지의 경우 공모 행위 등 범죄 혐의 입증에 결정적인 증거가 될 가능성이 크다. 그러나 현재까지 협업 툴과 관련된 연구는 모바일 기기에서 메시지를 획득하는 것에 초점이 맞추어져, 윈도우 환경의 업무용 PC에서 메시지를 획득하는 방법에 대한 연구는 미흡하다. 따라서 윈도우 환경에서 설치형 어플리케이션 형태로 동작하는 협업 툴을 중심으로 메시지를 획득하는 방안에 대한 연구가 필요하다.

본 논문에서는 윈도우 환경에서 설치형 어플리케이션을 통해 협업 툴을 사용하였을 때 메시지 데이터가 저장되는 파일을 식별하고, 해당 파일의 데이터 구조를 분석하여 사용자가 송·수신한 메시지를 획득하였다. 논문의 구성은 다음과 같다. 2장에서는 기존 연구와 한계점을 설명하고, 3장에서는 설치형 어플리케이션 Electron App의 특성을 토대로 연구방법을 정리한다. 4장에서는 잔디(jandi), 슬랙(slack), 팀즈(microsoft teams) 각 협업 툴 별로 메시지 데이터가 저장된 파일을 분석하여 메시지를 획득하고 활용 방안을 제시한다. 마지막 5장에서는 결론으로 마무리한다.

## II. 관련 연구

본 논문의 연구 주제와 관련된 선행연구 및 한계점은 다음과 같다. Slack 및 Discord 메신저에 관한 연구[2]와 협업 툴 아티팩트 분석 및 삭제된 데이터 복구 연구[3]에서는 윈도우 환경이 아닌 모바일 기기에서만 메시지를 확인하였다. Windows에서의 Wire 크리덴셜 획득 및 아티팩트 분석[4]의 경우 윈도우 환경에서 메시지 데이터 구조와 내용을 확인하였다. WhatsApp Web 버전 포렌식 조사[5]에서는 사용자 행위와 관련된 레코드를 발견하였으나 Web 버전만 연구가 진행되었다. 협업 툴의 사용자 행위별 아티팩트 분석 연구[6]는 윈도우 환경에서 팀즈의 메시지 데이터 구조 중 일부를 확인하였으나 메시지 획득 시 윈도우 환경이 아닌 모바일 기기의 데이터베이스 파일을 사용하였다.

한편, CCL Solutions[7][8][9]에서는 Electron App의 IndexedDB에 관해 연구하고 분석 도구를 개발하였으나, 본 논문의 연구 대상 협업 툴을 분석하는 과정에서 오류가 발생하거나 데이터를 온전히 획득하지 못하였다. Electron 기반 협업 프로그램의 취약점에 관한 연구[10]도 존재하나, 메시지 획득에 관한 연구는 진행되지 않았다.

본 논문에서는 윈도우 환경에서 협업 툴 잔디, 슬랙, 팀즈를 중심으로 연구를 진행하고, 이를 바탕으로 Electron App의 메시지를 획득하는 방법을 제시한다.

## III. 연구 방법

### 3.1 사전 지식

최근 다양한 기기에서의 접근성을 보장하는 크로스 플랫폼 어플리케이션의 경우 Electron 오픈소스 프레임워크를 사용하여 개발되고 있으며 이러한 어플리케이션을 Electron App이라 한다. 논문 작성 시점 기준으로 약 800여 개의 어플리케이션이 Electron App 형태로 개발되었고 협업 툴 잔디, 슬랙, 팀즈 또한 Electron App에 속한다[15]. Electron 프레임워크가 Chromium과 Node.js 기반의 Web 기술을 이용하므로, Electron App은 Fig. 1.과 같이 Chromium Web Browser와 동일하게 웹 브라우저 저장소(web storage)의 형태인 Cache, IndexedDB, Local Storage, Session Storage, Cookies에 데이터를 저장한다[11]. 프로그램별 파일 경로는 Table 1.과 같다.

Table 1. File Path of Electron App and Chromium Web Browser

| Type | Program | File Path |
|---|---|---|
| Electron App | Jandi | %AppData%\JANDI |
| | Slack | %AppData%\Slack |
| | Teams | %AppData%\Microsoft\Teams |
| Chromium Web Browser | Chrome | %LocalAppData%\Google\Chrome\User Data\Default |
| | Microsoft Edge | %LocalAppData%\Microsoft\Edge\User Data\Default |

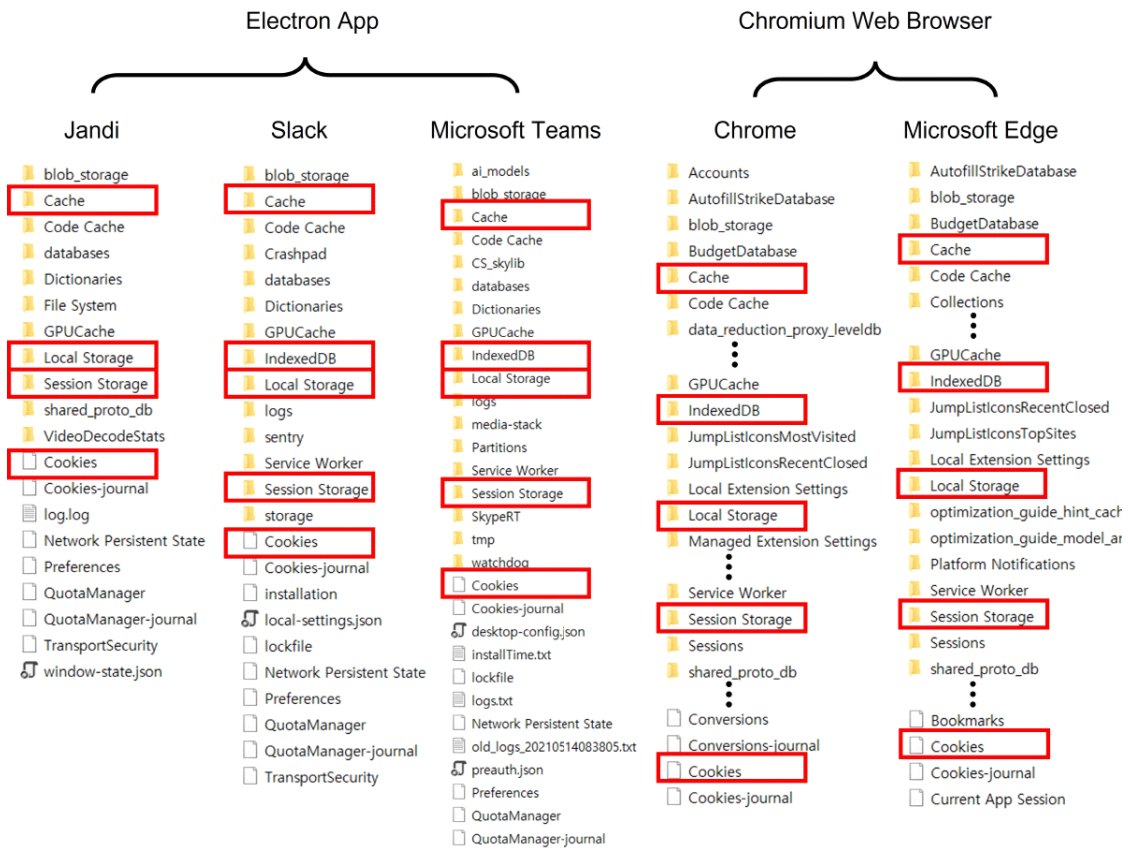



Fig. 1. Web Storage of Electron App and Chromium Web Browser

### 3.2 파일 식별 및 분석

앞서 살펴본 바와 같이 Electron App이 Chromium Web Browser와 데이터를 저장하는 방식이 동일하다는 점에 착안하여, Windows 10 운영체제에서 웹 브라우저 크롬, 마이크로소프트 엣지를 통해 협업 툴 잔디, 슬랙, 팀즈에 접속한 다음 Web Storage 중 메시지가 저장되는 파일을 식별하고 Electron App에서도 같은 유형의 파일을 분석하였다.

Fig. 2.에서 보는 것처럼 Web Storage에 저장된 데이터는 웹 개발자 도구(web developer tools)를 통해 확인하였고, Cache는 Chrome Cache View[12]를 이용하여 데이터를 추출한 후 확인하였다. 확인 결과는 Table 2.와 같으며 해당 파일들에 대해 HxD[14] 등을 사용하여 세부 데이터 구조를 분석하고 메시지 데이터를 획득하였다.

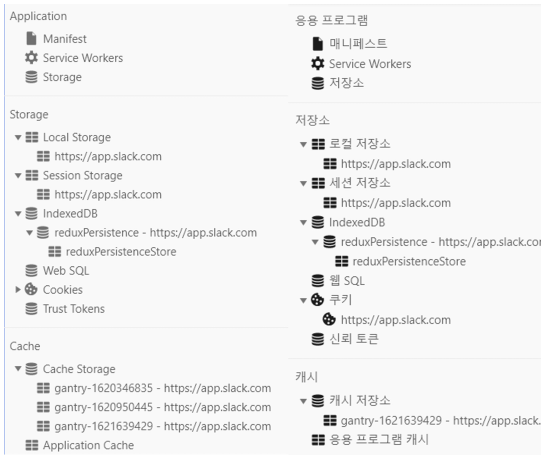



Fig. 2. Web Developer Tools of Chrome and Microsoft Edge

Table 2. File Path where Message Data are stored

| Tool | How to use | File Type | File Path |
|---|---|---|---|
| Jandi | Chromium Web Browser (Chrome) | Cache | %LocalAppData%\Google\Chrome\User Data\Default**Cache** |
| | Chromium Web Browser (MS Edge) | | %LocalAppData%\Microsoft\Edge\User Data\Default**Cache** |
| | Electron App | | %AppData%\JANDI**Cache** |
| Slack | Chromium Web Browser (Chrome) | IndexedDB (Blob) | %LocalAppData%\Google\Chrome\User Data\Default**IndexedDB**\https_app.slack.com_0.indexeddb.blob\1\00\[#] |
| | Chromium Web Browser (MS Edge) | | %LocalAppData%\Microsoft\Edge\User Data\Default**IndexedDB**\https_app.slack.com_0.indexeddb.blob\1\00\[#] |
| | Electron App | | %AppData%\Slack**IndexedDB**\https_app.slack.com_0.indexeddb.blob\1\00\[#] |
| Microsoft Teams | Chromium Web Browser (Chrome) | IndexedDB (LevelDB) | %LocalAppData%\Google\Chrome\User Data\Default**IndexedDB**\https_teams.live.com_0.indexeddb.leveldb\[#.log, #.ldb] |
| | Chromium Web Browser (MS Edge) | | %LocalAppData%\Microsoft\Edge\User Data\Default**IndexedDB**\https_teams.live.com_0.indexeddb.leveldb\[#.log, #.ldb] |
| | Electron App | | %AppData%\Microsoft\Teams**IndexedDB**\https_teams.live.com_0.indexeddb.leveldb\[#.log, #.ldb] |

# IV. 메시지 획득

## 4.1 잔디 (jandi)

### 4.1.1 team, room

잔디는 팀(team)과 대화방(room)으로 구성되며 대화방은 토픽, 채팅으로 구분된다. Fig. 3.에서 보는 것처럼 이용자는 팀별로 대화방을 생성하여 메시지, 파일 등을 송·수신한다.

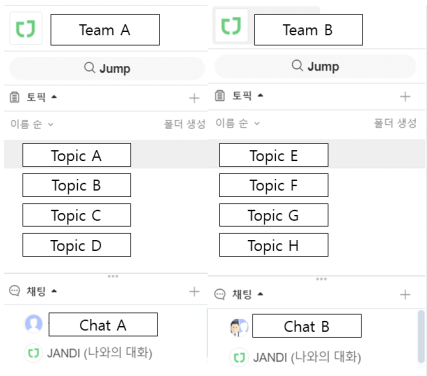
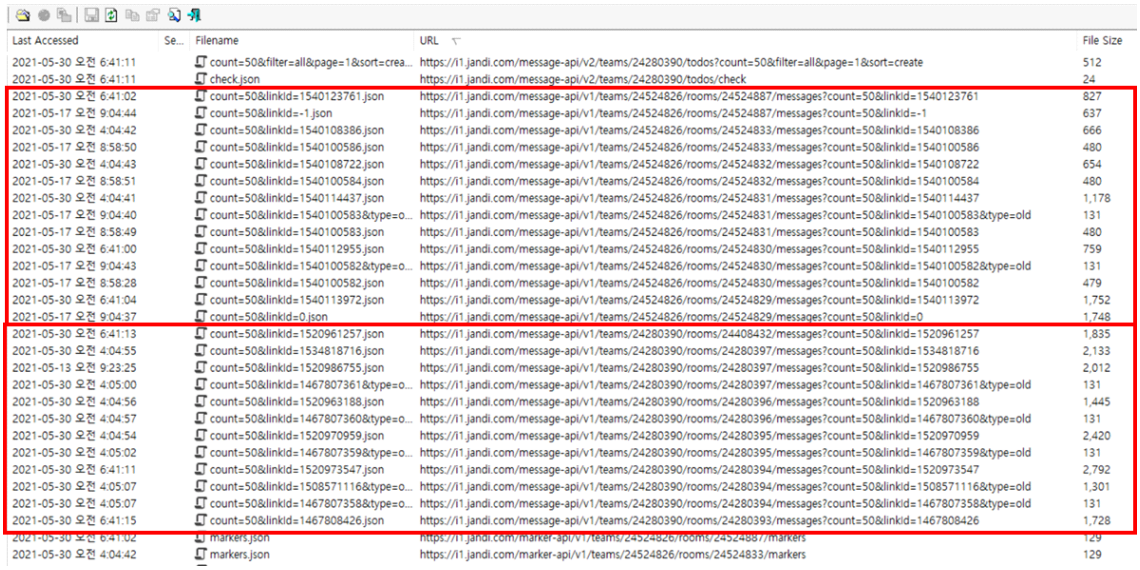



Fig. 3. team, room of Jandi

### 4.1.2 Cache

잔디는 캐시 파일에 메시지 데이터를 저장하며 파일 경로는 Table 2.와 같다. 캐시에 저장된 정보는 ChromeCacheView[12]를 이용하여 확인할 수 있고, 메시지 데이터의 경우 Fig. 4.에서 보는 것처럼 캐시 속성 중 URL 값이 "https://i1.jandi.com/message-api/v1/teams/[team id]/rooms/[room id]/messages?" 형태인 JSON 파일에 저장된다. team id는 사용자가 협업 툴 사용을 위해 생성한 팀의 고유 id 값이고, room id는 team 내부에 업무 협업을 위해 생성한 room에 대한 고유 id 값이다. team id, room id는 각각 생성된 팀 개수, 팀 내 대화방 개수만큼 생성된다.

Fig. 3.과 같은 테스트 환경에서 팀 이름과 대화방 이름을 매핑한 결과는 Table 3.와 같다. Team A와 Team B는 각각 6개의 room id 값을 갖고 있으며 캐시 속성 중 URL에 team id와 room id 값을 매핑하여 상호간의 메시지 전달 그룹을 구분한다.



| Last Accessed | Se... | Filename | URL | File Size |
|---|---|---|---|---|
| 2021-05-30 오전 6:41:11 | | count=50&filter=all&page=1&sort=crea... | https://i1.jandi.com/message-api/v2/teams/24280390/todos?count=50&filter=all&page=1&sort=create | 512 |
| 2021-05-30 오전 6:41:11 | | check.json | https://i1.jandi.com/message-api/v2/teams/24280390/todos/check | 24 |
| 2021-05-30 오전 6:41:02 | | count=50&linkId=1540123761.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524887/messages?count=50&linkId=1540123761 | 827 |
| 2021-05-17 오전 9:04:44 | | count=50&linkId=-1.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524887/messages?count=50&linkId=-1 | 637 |
| 2021-05-30 오전 4:04:42 | | count=50&linkId=1540108386.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524833/messages?count=50&linkId=1540108386 | 666 |
| 2021-05-17 오전 8:58:50 | | count=50&linkId=1540100586.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524833/messages?count=50&linkId=1540100586 | 480 |
| 2021-05-30 오전 4:04:43 | | count=50&linkId=1540108722.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524832/messages?count=50&linkId=1540108722 | 654 |
| 2021-05-17 오전 8:58:51 | | count=50&linkId=1540100584.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524832/messages?count=50&linkId=1540100584 | 480 |
| 2021-05-30 오전 4:04:41 | | count=50&linkId=1540114437.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524831/messages?count=50&linkId=1540114437 | 1,178 |
| 2021-05-17 오전 9:04:40 | | count=50&linkId=1540100583&type=o... | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524831/messages?count=50&linkId=1540100583&type=old | 131 |
| 2021-05-17 오전 8:58:49 | | count=50&linkId=1540100583.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524831/messages?count=50&linkId=1540100583 | 480 |
| 2021-05-30 오전 6:41:00 | | count=50&linkId=1540112955.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524830/messages?count=50&linkId=1540112955 | 759 |
| 2021-05-17 오전 9:04:43 | | count=50&linkId=1540100582&type=o... | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524830/messages?count=50&linkId=1540100582&type=old | 131 |
| 2021-05-17 오전 8:58:28 | | count=50&linkId=1540100582.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524830/messages?count=50&linkId=1540100582 | 479 |
| 2021-05-30 오전 6:41:04 | | count=50&linkId=1540113972.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524829/messages?count=50&linkId=1540113972 | 1,752 |
| 2021-05-17 오전 9:04:37 | | count=50&linkId=0.json | https://i1.jandi.com/message-api/v1/teams/24524826/rooms/24524829/messages?count=50&linkId=0 | 1,748 |
| 2021-05-30 오전 6:41:13 | | count=50&linkId=1520961257.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24408432/messages?count=50&linkId=1520961257 | 1,835 |
| 2021-05-30 오전 4:04:55 | | count=50&linkId=1534818716.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280397/messages?count=50&linkId=1534818716 | 2,133 |
| 2021-05-13 오전 9:23:25 | | count=50&linkId=1520986755.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280397/messages?count=50&linkId=1520986755 | 2,012 |
| 2021-05-30 오전 4:05:00 | | count=50&linkId=1467807361&type=o... | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280397/messages?count=50&linkId=1467807361&type=old | 131 |
| 2021-05-30 오전 4:04:56 | | count=50&linkId=1520963188.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280396/messages?count=50&linkId=1520963188 | 1,445 |
| 2021-05-30 오전 4:04:57 | | count=50&linkId=1467807360&type=o... | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280396/messages?count=50&linkId=1467807360&type=old | 131 |
| 2021-05-30 오전 4:04:54 | | count=50&linkId=1520970959.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280395/messages?count=50&linkId=1520970959 | 2,420 |
| 2021-05-30 오전 4:05:02 | | count=50&linkId=1467807359&type=o... | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280395/messages?count=50&linkId=1467807359&type=old | 131 |
| 2021-05-30 오전 6:41:11 | | count=50&linkId=1520973547.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280394/messages?count=50&linkId=1520973547 | 2,792 |
| 2021-05-30 오전 4:05:07 | | count=50&linkId=1508571116&type=o... | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280394/messages?count=50&linkId=1508571116&type=old | 1,301 |
| 2021-05-30 오전 4:05:07 | | count=50&linkId=1467807358&type=o... | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280394/messages?count=50&linkId=1467807358&type=old | 131 |
| 2021-05-30 오전 6:41:15 | | count=50&linkId=1467808426.json | https://i1.jandi.com/message-api/v1/teams/24280390/rooms/24280393/messages?count=50&linkId=1467808426 | 1,728 |
| 2021-05-30 오전 6:41:02 | | markers.json | https://i1.jandi.com/marker-api/v1/teams/24524826/rooms/24524887/markers | 129 |
| 2021-05-30 오전 4:04:42 | | markers.json | https://i1.jandi.com/marker-api/v1/teams/24524826/rooms/24524833/markers | 129 |

Fig. 4. Jandi Cache with ChromeCacheView

Table 3. URL Format of JSON files in Jandi Cache

| team | room | URL | | | | |
|---|---|---|---|---|---|---|
| | | common | team id | common | room id | common |
| Team A | Chat A | https://i1.jandi.com /message-api/v1/teams/ | 24524826 | /rooms/ | 24524887 | /messages? |
| | Topic D | | | | 24524833 | |
| | Topic C | | | | 24524832 | |
| | Topic B | | | | 24524831 | |
| | Topic A | | | | 24524830 | |
| | Chat(JANDI) | | | | 24524829 | |
| Team B | Chat B | | 24280390 | | 24408432 | |
| | Topic H | | | | 24280397 | |
| | Topic G | | | | 24280396 | |
| | Topic F | | | | 24280395 | |
| | Topic E | | | | 24280394 | |
| | Chat(JANDI) | | | | 24280393 | |

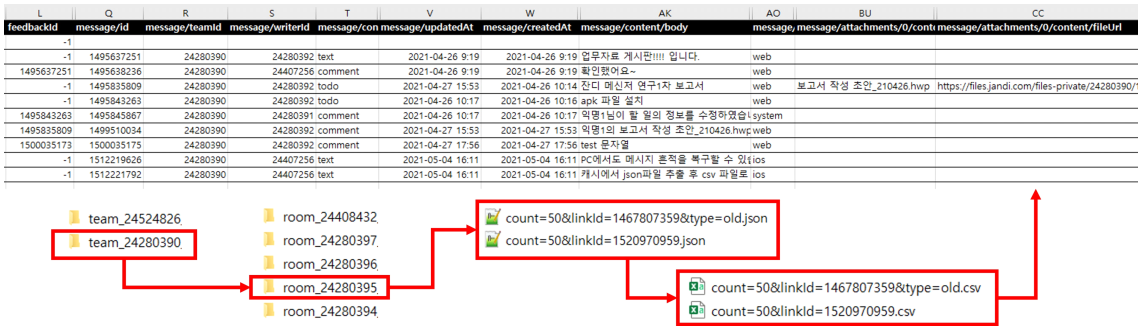



Fig. 5. Message Acquisition Process of Jandi

#### 4.1.3 메시지 획득 방안

잔디에서 메시지를 획득하는 과정은 다음과 같다. ❶ Fig. 4.와 같이 ChromeCacheView를 통해 잔디의 캐시 정보를 확인하고 URL 속성으로 정렬한다. ❷ Table 3.의 URL Format을 참고하여 Fig. 5.에 도식된 것처럼 캐시 내 JSON 파일을 team id, room id를 구분하여 추출한다. ❸ 추출된 JSON 파일은 JSON To CSV Converter[13]를 이용하여 CSV 형태로 변환한다. ❹ 엑셀 프로그램을 통해 CSV 파일을 로드한다. ❺ 로드된 데이터들을 순차적으로 병합하고 날짜 필드로 정렬하면 Fig. 5.와 같이 팀별로 특정 대화방에서 송·수신한 메시지 내용, 첨부파일명, 작성자, 작성시각 등을 확인할 수 있다. 단, 직접적인 팀명, 대화방명, 작성자명은 확인할 수 없으며 그에 대응되는 고유 식별자인 teamId, roomId, writerId 정보만이 확인 가능하다. 또한, fileUrl을 통한 첨부파일의 직접 접근도 제한된다.

### 4.2 슬랙(Slack)

#### 4.2.1 workspace, channel

슬랙은 워크스페이스(workspace)와 채널(channel)로 구성되며 DM(Direct Message)은 채널 중 하나이다. 슬랙의 워크스페이스, 채널, DM은 각각 잔디의 팀, 토픽, 채팅에 대응되며 같은 기능을 담당한다.

#### 4.2.2 IndexedDB

슬랙은 IndexedDB 중 Blob 파일에 메시지 데이터를 저장하며 파일 경로는 Table 2.와 같다. 웹 개발자 도구(web developer tools)를 이용하여 웹 브라우저의 IndexedDB에 저장된 정보를 확인할 수 있고, Fig. 7.에서 보는 것처럼 IndexedDB 하위 reduxPersistenceStore에서 메시지를 확인하였다. 메시지는 workspace id와 channel id를 구분자로 하여 워크스페이스 내 채널 별로 저장된다. workspace id와 channel id는 웹 브라우저를 통해 슬랙 접속 시 URL에서 "https://app.slack.com/client/[workdspace id]/[channel id]"의 형태로 확인할 수 있다.

#### 4.2.3 메시지 속성

슬랙의 메시지 속성 정보는 Fig. 7.과 같이 웹 개발자 도구를 통해 확인 가능하고, 속성 간 순서는 Fig. 6.와 같이 HxD[14]를 통해 Blob 파일의 Hex 값을 분석하여 파악할 수 있다. 슬랙의 메시지 속성을 정리하면 Table 4.와 같으며 중요 속성 정보는 음영처리 하였다. 메시지 속성 중 ts, thread_ts 속성은 Unix Timestamp 형태로 저장되며, ts 속성은 메시지 작성 시간을, thread_ts 속성은 메시지가 댓글에 해당할 경우 원글의 작성 시간을 의미한다. text 속성에는 메시지 내용이 저장되며 영문 메시지는 ASCII, 한글 메시지는 UTF-16으로 인코딩되어 저장된다. files, channel, user 속성에는 각각 파일, 채널, 작성자의 고유 식별자가 저장된다.

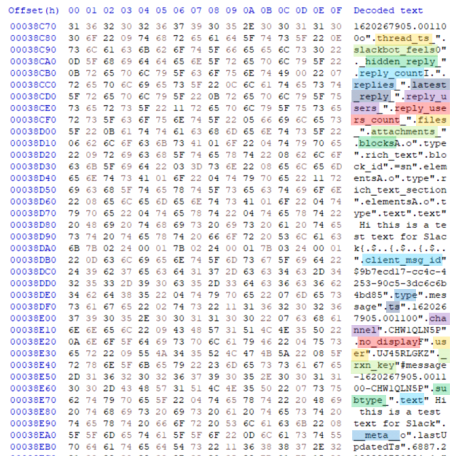



Fig. 6. Hex Data of Slack Message

Table 4. Slack Message Attribute

| # | Attribute | # | Attribute |
|---|---|---|---|
| 1 | thread_ts | 12 | client_msg_id |
| 2 | slackbot_feels | 13 | type |
| 3 | _hidden_reply | 14 | ts |
| 4 | reply_count | 15 | channel |
| 5 | replies | 16 | no_display |
| 6 | latest_reply | 17 | user |
| 7 | reply_users | 18 | _rxn_key |
| 8 | reply_users_count | 29 | subtype |
| 9 | files | 20 | text |
| 10 | attachments | 21 | __meta__ |
| 11 | blocks | | |

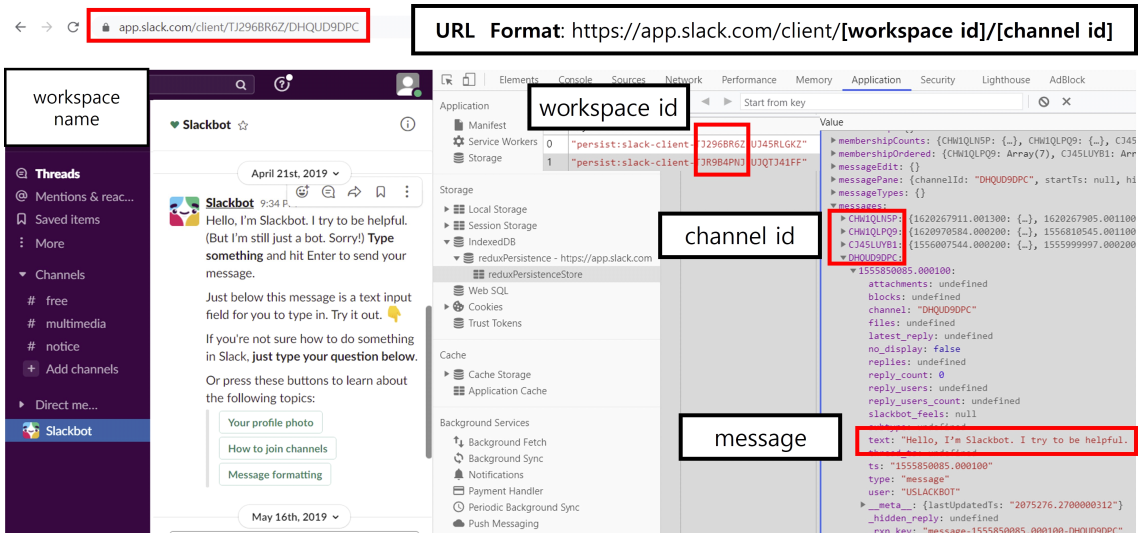



Fig. 7. Slack with Web Developer Tools

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| team_id | team_name | | | | | |
| T027WLKGWQL | TEAM A | | | | | |
| | | | | | | |
| member_id | member_name | member_real_name | | | | |
| U027HN15VF0 | abcd1234 | John | | | | |
| U02835LJTDX | aaaa2021 | Peter | | | | |
| U027HRD1CVC | bbbbcc | James | | | | |
| USLACKBOT | slackbot | Slackbot | | | | |
| | | | | | | |
| channel_id | channel_name | | | | | |
| C027HN16B8W | Channel A | | | | | |
| C027HN2F2KG | Channel B | | | | | |
| C027HNDSEHL | Channel C | | | | | |
| | | | | | | |
| file_id | file_name | | | | | |
| F028UBUTF98 | Attachment_A.zip | | | | | |
| F028J3TNW2H | Attachment_B.pdf | | | | | |
| F029AQNL5AL | Attachment_C.zip | | | | | |
| F0290G6R8FP | image.jpg | | | | | |
| | | | | | | |
| thread_ts | files | ts | channel | user | text | time |
| _ | F028UBUTF98 | 1626919953.000300 | C027HNDSEHL | U02835LJTDX | 저도 추가 자료 보내드려요 | 2021-07-22 11:12 |
| _ | F0286LTDBHV | 1626790765.001900 | C027HNDSEHL | U027HRD1CVC | 관련자료 송부드립니다. | 2021-07-20 23:19 |
| _ | _ | 1626787303.001500 | C027HNDSEHL | U027HN15VF0 | 전산학 필기 내 *기출문제* 같이 | 2021-07-20 22:21 |
| _ | F028J3TNW2H | 1626787247.000600 | C027HNDSEHL | U027HN15VF0 | | 2021-07-20 22:20 |
| _ | F029AQNL5AL | 1626787208.000300 | C027HNDSEHL | U027HN15VF0 | | 2021-07-20 22:20 |
| _ | _ | 1626060588.001000 | C027HNDSEHL | U027HRD1CVC | <@U027HRD1CVC> 님이 채널에 | 2021-07-12 12:29 |

Fig. 8. Slack Message Acquisition through a Developed Carving Tool

#### 4.2.4 메시지 획득 방안

슬랙의 메시지는 메시지 속성 정보를 기반으로 카빙(carving) 기법을 통해 획득한다. IndexedDB의 Blob 파일을 대상으로 Hex 값 단위에서 메시지 데이터와 동일한 패턴을 보이는 데이터를 추출하고 메시지 속성 정보로 데이터를 해석하면 메시지 내용을 획득할 수 있다.

본 논문에서는 위의 과정을 자동화한 Python v3.7 기반의 메시지 카빙 도구[1]를 개발하였다. 개발 과정은 다음과 같다. ❶ Table 4.의 메시지 속성에 대한 정규표현식을 사용하여 메시지 데이터 패턴과 동일한 패턴을 보이는 데이터를 탐색한다. ❷ 데이터 탐색 후에는 Table 4.에서 음영처리한 주요 메시지

1) The Message Carving Tool for Slack and Microsoft Teams, https://github.com/c0msherl0ck/CollaborationTool

속성 thread_ts, files, ts, channel, user, text에 해당하는 값들에 대해서만 데이터를 추출한다. ❸ Unix Timestamp 형식으로 저장된 ts, thread_ts 속성에 대해서는 Date 변환함수를 사용하여 시간 정보를 획득하고, text 속성에 저장된 메시지의 경우 영문은 ASCII, 한글은 UTF-16으로 디코딩하여 해석한다. ❹ 한편, 메시지 데이터 이외에도 files, user, channel 속성에는 고유 식별자가 저장되는데, 이에 대응되는 정보를 얻기 위한 데이터를 추가로 획득한다. 획득 방법은 앞서 기술한 메시지 데이터를 획득하는 방법과 동일하며 탐색을 위한 속성 정보만이 다르다. ❺ 메시지 내용을 비롯하여 획득한 데이터는 CSV 파일에 순차적으로 행 단위로 저장한다. ❻ 엑셀 프로그램으로 CSV 파일을 로드할 경우 Fig. 8.과 같이 팀, 사용자, 채널, 첨부파일, 메시지 관련 정보를 확인할 수 있다.

## 4.3 팀즈 (Microsoft Teams)

### 4.3.1 conversation

팀즈는 전화번호를 인증한 하나의 계정당 복수의 대화방(conversation)을 생성할 수 있으며 대화방에서 메시지, 파일 등을 송·수신한다.

### 4.3.2 IndexedDB

팀즈는 IndexedDB 중 LevelDB 파일에 메시지 데이터를 저장하며 파일 경로는 Table 2.와 같다. 웹 개발자 도구(web developer tools)를 이용하여 웹 브라우저의 IndexedDB에 저장된 정보를 확인할 수 있고, Fig. 9.에서 보는 것처럼 IndexedDB 하위 replychains에서 메시지 내용을 확인하였다. 메시지는 conversation id를 구분자로 하여 대화방별로 저장되며, conversation id는 웹 브라우저를 통해 팀즈 접속 시 URL에서 "https://teams.live.com/_#/conversations/{conversation id}?"의 형태로 확인할 수 있다.

### 4.3.3 메시지 속성

팀즈의 메시지 속성 정보는 Fig. 9.와 같이 웹 개발자 도구를 통해 확인 가능하고, 속성 간 순서는 Fig. 10.와 같이 HxD[14]를 통해 LevelDB 파일의 Hex 값을 분석하여 파악할 수 있다. 메시지 속성은 Table 5.와 같으며 메시지 분석에 있어 중요한 속성은 음영처리 하였다. 메시지가 전달되는 대화방은 conversationId를 기준으로 구분되고, 메시지는 clientmessageid를 고유 식별자로 가진다. 메시지 내용과 첨부파일은 content, attachments 속성에 저장되고, 작성자의 이름과 사용자 ID는 imdisplayname과 creator 속성에 저장된다. 시간 정보는 3가지 속성 composetime, originalarrivaltime, clientArrivalTime에 저장되며 각각 발신, 수신, 사용자 확인 시간을 의미한다.

### 4.3.4 메시지 획득 방안

팀즈의 메시지는 슬랙과 마찬가지로 메시지 속성 정보를 기반으로 카빙(carving) 기법을 통해 획득한다. IndexedDB의 LevelDB 파일을 대상으로 Hex 값 단위에서 메시지 데이터와 동일한 패턴을 보이는 데이터를 추출하고 속성 정보에 따라 데이터를 해석하면 메시지 내용을 획득할 수 있다.

본 논문에서는 위의 과정을 자동화한 Python v3.7 기반의 메시지 카빙 도구[2]를 개발하였다. 개발 과정은 다음과 같다. ❶ Table 5.의 메시지 속성에 대한 정규표현식을 사용하여 메시지 데이터 패턴과 동일한 패턴을 보이는 데이터를 탐색한다. ❷ 데이터 탐색 후에는 Table 5.에서 음영처리한 주요 메시지 속성 content, clientmessageid, imdisplayname, composetime, originalarrivaltime, clientArrivalTime, conversationId, parentMessageId, creator, attachments에 해당하는 값들에 대해서만 데이터를 추출한다. ❸ text 속성에 저장된 메시지의 경우 영문은 ASCII, 한글은 UTF-16으로 디코딩하여 해석하고, attachments 속성에 저장된 데이터의 경우 하위 속성 objectId, itemId, title에 따라 재해석한다. ❹ 메시지 내용을 비롯하여 획득한 데이터는 CSV 파일에 순차적으로 행 단위로 저장한다. ❺ 엑셀 프로그램으로 CSV 파일을 로드하면 Fig. 11.에서 보는 것처럼 메시지 내용, 작성자, 수발신 시간, 대화방, 첨부파일명 등의 정보를 확인할 수 있다.

2) The Message Carving Tool for Slack and Microsoft Teams, https://github.com/c0msherl0ck/CollaborationTool

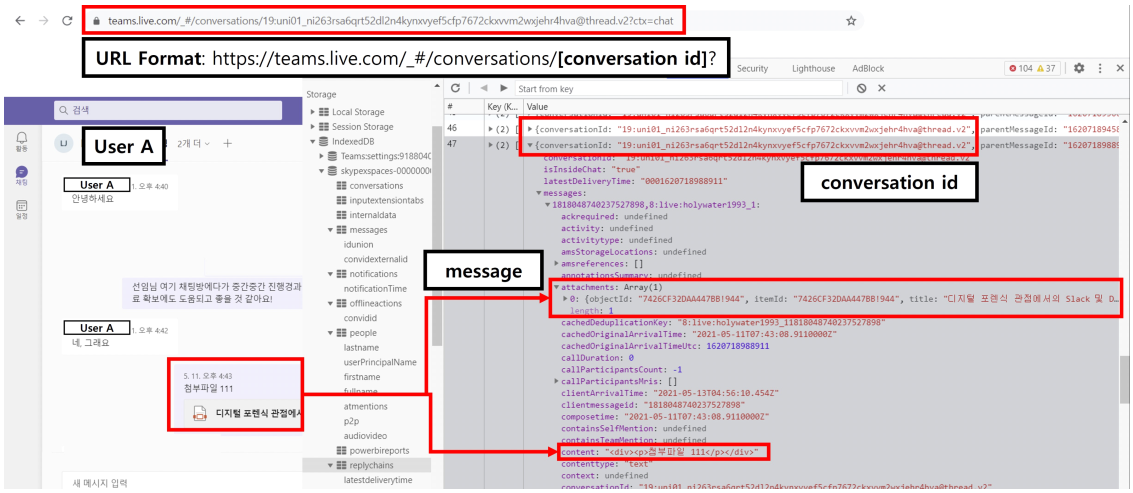



Fig. 9. Microsoft Teams with Web Developer Tools

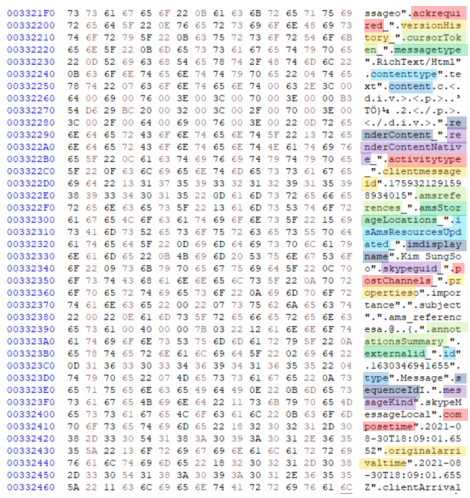


Fig. 10. Hex Data of Microsoft Teams Message

| content | clientmessageid | imdisplayname | composetime | originalarrivaltime | clientArrivalTime | conversationId | parentMessageId | creator | attachments |
|---|---|---|---|---|---|---|---|---|---|
| 넵!! | 3.26786E+18 | User A name | 2021-05-31T09:30:44 | 2021-05-31T09:30:44.[illegible] | [illegible] | [illegible] | [illegible] | [illegible] | |
| @@@@@@ test 문자열 @@@@@ < | 1.04003E+18 | User A name | 2021-05-07T01:27:25 | 2021-05-07T01:27:25.[illegible] | [illegible] | [illegible] | | | |
| | 0 | | 0 | 0 | [illegible] | [illegible] | Deleted Message | | |
| | 0 | | 0 | 0 | [illegible] | [illegible] | [illegible] | [illegible] | |
| <addmember><eventtime>1620718837214</eventtir | [illegible] | | 2021-05-11T07:40:37 | 2021-05-11T07:40:37.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | G19:uni01_ni263rsa6qrt52dl2n4kynxvyef5cfp7672ck | | |
| 안녕하세요 | 1.62072E+12 | User B name | 2021-05-11T07:40:37 | 2021-05-11T07:40:37.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| | 2.94915E+18 | User A name | 2021-05-11T07:41:03 | 2021-05-11T07:41:03.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | |
| 선임님 여기 채팅방에다가 중간중간 진행경. | 2.30333E+18 | User A name | 2021-05-11T07:42:16 | 2021-05-11T07:42:16.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | |
| 네, 그래요 | 1.62072E+12 | User B name | 2021-05-11T07:42:25 | 2021-05-11T07:42:25.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| 첨부파일 111 | 1.81805E+18 | User A name | 2021-05-11T07:43:08 | 2021-05-11T07:43:08.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | 디지털 포렌식 관점에서의 Slack 및 Dis |
| 검사역님, 논문 작성한게 있나요? | 1.62072E+12 | User B name | 2021-05-11T07:53:39 | 2021-05-11T07:53:39.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| 아니요 | 1.62072E+12 | User B name | 2021-05-11T07:54:17 | 2021-05-11T07:54:17.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| 지금 실험하시는거요 | 1.62072E+12 | User B name | 2021-05-11T07:54:23 | 2021-05-11T07:54:23.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| 아직이요..! | 2.70478E+17 | User A name | 2021-05-11T07:54:37 | 2021-05-11T07:54:37.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | |
| 제가 21일날 해외 컨퍼런스 발표를 하나해요</d | 1.62072E+12 | User B name | 2021-05-11T07:55:10 | 2021-05-11T07:55:10.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |
| 아 그때 말씀하시는거죠! | 1.58141E+18 | User A name | 2021-05-11T07:55:24 | 2021-05-11T07:55:24.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | |
| 네네 | 4.30109E+18 | User A name | 2021-05-11T07:55:31 | 2021-05-11T07:55:31.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User A ID | | |
| 그래서 그때 지나고 나서나 써도 쓸거 같은데요< | 1.62072E+12 | User B name | 2021-05-11T07:55:34 | 2021-05-11T07:55:34.2021-08-30T18:07:21.G19:uni01_ni263rsa6c | | 1.62072E+12 | User B ID | | |

Fig. 11. Microsoft Teams Message Acquisition through a Developed Carving Tool

Table 5. Microsoft Teams Message Attribute

| # | Attribute | # | Attribute | # | Attribute |
|---|---|---|---|---|---|
| 1 | ackrequired | 34 | messageStorageState | 67 | isPlainTextConvertedToHtml |
| 2 | versionHistory | 35 | isActionExecuteUpdate | 68 | isRichContentProcessed |
| 3 | cursorToken | 36 | _conversationIdMessageIdUnion | 69 | isRichMessagePropertiesProcessed |
| 4 | messagetype | 37 | parentMessageId | 70 | isRenderContentWithGiphyDisplayEnabled |
| 5 | contenttype | 38 | createdTime | 71 | isForceDelete |
| 6 | content | 39 | creator | 72 | isEditClientLie |
| 7 | renderContent | 40 | creatorProfile | 73 | reactionLieData |
| 8 | renderContentNative | 41 | isFromMe | 74 | originalNonLieMessage |
| 9 | activitytype | 42 | userHasStarred | 75 | originalNonLieReactions |
| 10 | clientmessageid | 43 | activity | 76 | originalNonLieReactionsSummary |
| 11 | amsreferences | 44 | previewData | 77 | context |
| 12 | amsStorageLocations | 45 | replyChainLatestDeliveryTime | 78 | isSfBGroupConversation |
| 13 | isAmsResourcesUpdated | 46 | scenarioName | 79 | convoCallId |
| 14 | imdisplayname | 47 | state | 80 | convoCallUrl |
| 15 | skypeguid | 48 | hydratedContent | 81 | eventId |
| 16 | postChannels | 49 | hydratedProperties | 82 | translation |
| 17 | properties | 50 | notificationLevel | 83 | dlpData |
| 18 | annotationsSummary | 51 | hasMessageActionFailed | 84 | layoutMetadata |
| 19 | externalid | 52 | messageSendErrorReason | 85 | messageLayoutType |
| 20 | id | 53 | messageSendDiagnosticError | 86 | callDuration |
| 21 | type | 54 | mentions | 87 | callParticipantsMris |
| 22 | sequenceId | 55 | hyperLinks | 88 | cachedDeduplicationKey |
| 23 | messageKind | 56 | attachments {objectId, itemId, title, ...} | 89 | cachedOriginalArrivalTime |
| 24 | composetime | 57 | inputExtensionAttachments | 90 | cachedOriginalArrivalTimeUtc |
| 25 | originalarrivaltime | 58 | processedInputExtensionAttachments | 91 | _emailDetails |
| 26 | clientArrivalTime | 59 | inlineImages | 92 | _callRecording |
| 27 | conversationLink | 60 | containsSelfMention | 93 | _callTranscript |
| 28 | from | 61 | containsTeamMention | 94 | _meetingObjects |
| 29 | source | 62 | teamMentionDisplayName | 95 | callParticipantsCount |
| 30 | idUnion | 63 | trimmedMessageContent | 96 | _pinState |
| 31 | conversationId | 64 | messageContentContainsImage | 97 | pinnedTime |
| 32 | versionNumber | 65 | messageContentContainsVideo | 98 | _policyViolation |
| 33 | version | 66 | isSanitized | 99 | _invalidateInvokeCacheDetails |

### 4.4 메시지 삭제

본 논문에서 메시지 삭제를 실험한 결과 잔디, 슬랙, 팀즈 세 협업툴 모두 메시지 삭제 이후 복구가 어려운 것으로 확인하였다. 잔디, 슬랙은 삭제 이후 삭제된 메시지와 관련된 흔적을 발견할 수 없었으며, 팀즈의 경우 Fig. 11.에서 보는 것처럼 메시지 속성 관련 정보는 남아있었으나 속성에 저장되는 데이터가 초기화되어 삭제 전 메시지를 확인할 수 없었다. 그러나, 메시지 삭제를 하지 않은 사용자가 다른 사용자의 삭제 행위 이후 협업 툴에 접속하지 않는다면, 해당 삭제 내역이 반영되지 않으므로 삭제 전 메시지를 확인할 수 있었다.

### 4.5 활용 시나리오

본 항에서는 시나리오를 바탕으로 연구결과를 활용할 수 있는 방안을 제시한다. 실제 발생 가능한 상황을 가정하여 모바일 기기 압수 제한, 메시지 삭제, 비밀번호 제출 거부 3가지 시나리오로 구성하였다.

#### 4.5.1 모바일 기기 압수 제한

(사건 개요)

A사에서 기업형 범죄가 발생하여 동사 사무실을 압수수색 중이다. 혐의자들은 범죄공모에 협업 툴 잔디를 사용하였고, 영장에 기재된 압수수색의 범위는 혐의자들의 업무용 PC로 제한되었다.

(증거 획득)

피압수자의 참여하에 동의를 얻어 압수 대상 업무용 PC에 로그인하고 Table 2.의 경로에 있는 Cache 파일을 압수한다. 이후, Chrome Cache View[12], JSON TO CSV Converter[13]를 사용하여 메시지를 획득하고, 범죄 관련성이 있는 내용만을 최종적으로 선별한다.

#### 4.5.2 메시지 삭제

(사건 개요)

B사는 협업 툴 슬랙을 도입하여 운용하고 있으며 임직원들은 PC, 모바일 기기 모두에서 슬랙을 이용하고 있다. B사를 압수수색 하던 중 현장에 나타나지 않던 혐의자 중 한 명이 범죄 행위를 은닉하고 증거를 인멸하기 위해 자신의 모바일 기기를 이용하여 슬랙에 있는 메시지를 삭제하였다.

(증거 획득)

압수수색 시작과 동시에 압수 대상 PC들의 네트워크를 차단하여 원격 조작을 방지한다. 이후, Table 2.의 경로에 있는 IndexedDB 파일을 압수하고, 본 논문에서 개발한 메시지 카빙 도구를 이용하여 삭제 행위 발생 전 메시지를 획득한다.

#### 4.5.3 비밀번호 제출 거부

(사건 개요)

협업 툴 팀즈를 사용한 혐의자들의 PC를 압수수색 중이다. 그러나 혐의자 중 일부는 PC 비밀번호 제출을 거부하고 있다. PC에 디스크 암호화 등의 보안정책은 설정되어 있지 않다.

(증거 획득)

Windows To Go[3] 방식으로 혐의자의 PC를 부팅하고 Table 2.의 경로에 있는 IndexedDB 파일을 압수한다. 이후, 본 논문에서 개발한 메시지 카빙 도구를 이용하여 메시지를 획득하고, 범죄 관련성이 있는 내용만을 최종 선별한다.

## V. 결 론

본 논문에서는 윈도우 환경에서 협업 툴 잔디, 슬랙, 팀즈를 중심으로 설치형 어플리케이션인 Electron App의 메시지를 획득하는 방안에 대해 연구하였다. Electron 오픈소스 프레임워크가 Web 기술을 기반으로 하므로, Electron App의 경우 웹 브라우저 저장소(web storage)와 같은 형태로 데이터를 저장한다는 점에 착안하여 메시지 데이터가 저장되는 파일을 식별하였다. 잔디는 Cache 파일에 슬랙과 팀즈는 IndexedDB 파일에 메시지를 저장하였으며, 해당 파일들에 대한 상세 분석 과정을 거쳐 메시지를 획득하였다. 메시지 획득 과정에서 Cache는 기존에 개발된 분석 도구를 활용하였고 IndexedDB는 본 논문에서 개발한 메시지 카빙

3) Windows To Go는 PC의 USB로 연결된 외부 드라이브에서 부팅할 수 있는 작업 영역을 만드는 데 사용되는 Windows의 기능이다.

도구를 이용하였다. 본 논문에서 제시한 방법을 활용할 경우 향후 PC 압수수색 과정에서 주요 증거인 메시지를 획득하는 데 도움이 될 것으로 기대한다.

## References


[1] Chosun Biz, https://biz.chosun.com/it-science/ict/2021/08/23/XCM6EEF6S5EJ5EUA4OJ6IVP62Q/, accessed Sep. 2021.

[2] Sumin Shin, Eunhu Park, Soram Kim, and Jongsung Kim, "Artifacts Analysis of Slack and Discord Messenger in Digital Forensic," Journal of Digital Contents Society, 21(4), pp. 799-809, Apr. 2020.

[3] Sumin Shin, Yongcheol Choi, Soram Kim, and Jongsung Kim, "Artifacts Analysis and Data Recovery of Collaboration Tools," Journal of Digital Forensics, 15(2), pp. 99-123, June. 2021.

[4] Sumin Shin, Soram Kim, Byungchul Youn, and Jongsung Kim, "Acquiring Credential and Analyzing Artifacts of Wire Messenger on Windows," Journal of The Korea Institute of information Security & Cryptology, 31(1), pp. 61-71, Feb. 2021.

[5] Furkan Paligu and Cihan Varo, "Browser Forensic Investigations of WhatsAppWeb Utilizing IndexedDB Persistent Storage," Future Internet, 12(11), Dec. 2020.

[6] Young-hoon Kim and Tae-kyoung Kwon, "On Artifact Analysis for User Behaviors in Collaboration Tools - Using differential forensics for distinct operating environments," Journal of The Korea Institute of information Security & Cryptology, 31(3), pp. 353-363, Jun. 2021.

[7] CCL Solutions Group, "IndexedDB on Chromium," https://www.cclsolutionsgroup.com/post/indexeddb-on-chromium, accessed Sep. 2021.

[8] CCL Solutions Group, "ccl_chrome_indexeddb," https://github.com/cclgroupltd/ccl_chrome_indexeddb, accessed Sep. 2021.

[9] CCL Solutions Group, "Hang on! That's not SQLite! Chrome, Electron and LevelDB," https://www.cclsolutionsgroup.com/post/hang-on-thats-not-sqlite-chrome-electron-and-leveldb, accessed Sep. 2021.

[10] Hyomin Lee, Yeonseok Jang, Yonghee Kwon, Eunji Lim, Jongmin Kim, and Jinwoo Park, "Analysis of Vulnerability in Electron Based Collaboration Tools," Journal of The Korea Institute of information Security & Cryptology, 31(4), pp. 573-586, Aug. 2021.

[11] web.dev, "Storage for theWeb," https://web.dev/storage-for-the-web, accessed Sep. 2021.

[12] Nirsoft, "Chrome Cache View v2.25," https://www.nirsoft.net/utils/chrome_cache_view.html, accessed Sep. 2021.

[13] Data Design Group, "Convert JSON to CSV," https://www.convertcsv.com/json-to-csv.htm, accessed Sep. 2021.

[14] mh-nexus, "HxD v2.5," https://mh-nexus.de/en/hxd/, accessed Sep. 2021.

[15] OpenJS Foundation, "Electron Apps," https://www.electronjs.org/apps, accessed Sep. 2021.

## 〈저자소개〉

김 성 수 (Sung-soo Kim) 정회원
2016년 2월: 고려대학교 컴퓨터학과 졸업
2021년 1월~현재: 금융감독원
〈관심분야〉 정보보호, 디지털 포렌식

이 성 진 (Sung-jin Lee) 종신회원
2004년 2월: 단국대학교 컴퓨터공학과 졸업
2006년 2월: 서울대학교 전기컴퓨터공학부 석사
2015년 8월: 연세대학교 법학 석사
2018년 2월: Indiana University Kelley School of Business MBA
2022년 2월: 성균관대학교 과학수사학과 박사
2011년 1월~현재: 금융감독원 자본시장특별사법경찰 선임검사역
〈관심분야〉 디지털포렌식, 전자금융, IT검사, 정보보호 등