Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.1, Feb. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.1.61

Windows에서의 Wire 크리덴셜 획득 및 아티팩트 분석*

신 수 민,[1†] 김 소 람,[1] 윤 병 철,[1] 김 종 성[2‡]
[1,2]국민대학교(대학원생, 교수)

Acquiring Credential and Analyzing Artifacts of Wire Messenger on Windows*

Sumin Shin,[1†] Soram Kim,[1] Byungchul Youn,[1] Jongsung Kim[2‡]
[1,2]Kookmin University (Graduate student, Professor)

요 약

인스턴트 메신저는 현대인의 의사소통 수단으로 스마트폰과 PC 등에서 이용 가능하며, 개별로 사용하거나 연동해서 사용할 수 있다. 메시지, 통화 및 파일 공유 등의 다양한 기능을 제공하는 메신저는 사용자의 행위정보를 포함하므로 포렌식 수사관점에서 중요한 증거로서의 가치를 지닌다. 하지만 스마트폰 데이터는 데이터 추출의 어려움이나 제작자의 보안기술 적용으로 분석의 어려움이 있다. 그러나 스마트폰과 PC 메신저를 연동해 사용하고 있는 경우 스마트폰 대신 PC에서 대화 내용 및 사용자 데이터 획득이 가능하다. 본 논문에서는 Windows 10 환경에서 Wire 메신저의 크리덴셜 정보를 획득했으며, 다른 PC에서 인증 과정 없이 로그인 가능함을 보였다. 또한, 사용자 행위를 기반으로 생성되는 주요 아티팩트 선별하고 정리하였다.

ABSTRACT

Instant messengers are a means of communication for modern people and can be used with smartphones and PCs respectively or connected with each other. Messengers, which provide various functions such as message, call, and file sharing, contain user behavior information regarded as important evidence in forensic investigation. However, it is difficult to analyze as well as acquire smartphone data because of the security of smartphones or apps. However, messenger data can be extracted through PC when the messenger is used on PC. In this paper, we obtained the credential data of Wire messenger in Windows 10, and showed that it is possible to log-in from another PC without authentication. In addition, we identified and classified major artifacts generated based on user behavior.

Keywords: Instant Messenger, Forensic Investigation, Credential, Artifact

I. 서 론

2020년, 신종 바이러스 감염증의 장기화로 외부 활동이 감소하고, 재택근무 및 원격 수업이 증가하면서 메신저 사용량이 급격하게 증가하고 있다[1]. 기존의 텍스트, 음성/영상 통화, 멀티미디어 파일 공유 기능뿐만 아니라 협업 기능 등을 추가로 제공하면서 양질의 서비스를 제공하고 있다. 이러한 메신저 데이터는 개인정보 및 사용자의 주요 행위정보를 포함하고 있으므로 포렌식 수사에서 중요한 증거로서의 가치를 갖는다. 실제로 메신저는 범죄 행위의 주요한 수단으로 활용되며, 2020년에도 텔레그램, Wire와 Discord 등의 메신저를 통해 성착취물을 제작하고 유포한 사건이 발생했다[2]. 그러나 이러한 메신저 데이터들을 증거로 활용하기 위해서는 선행 연구를 통해 명확한 데이터 분석 방향이 제시되어야 할 필요가 있다. 또한, 메신저 앱들은 스마트폰뿐만 아니라

Received(01. 11. 2021), Modified(02. 15. 2021), Accepted(02. 15. 2021)
* 본 연구는 2020년 대검찰청의 지원을 받아 수행한 연구용역 결과임
† 주저자, tnals523@kookmin.ac.kr
‡ 교신저자, jskim@kookmin.ac.kr(Corresponding author)

PC로 확장되어 동일한 기능을 제공하고 있으므로 다양한 환경에서의 데이터 획득 및 분석결과가 요구된다.

본 논문에서는 Windows 환경에서의 Wire 메신저를 대상으로 로컬 데이터 및 네트워크 패킷 분석을 통해 크리덴셜을 식별하고 및 활용방안을 제시하였으며, 주요 아티팩트 분석을 수행하였다. 계정 정보 및 대화 내역을 확인하였으며, 삭제된 데이터 복구 및 식별 방안을 연구하였다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 설명하고, 3장에서는 획득한 크리덴셜 정보를 설명하고 이에 대한 활용 가능 여부를 설명한다. 4장에서는 사용자 행위를 기반으로 생성된 아티팩트를 정리했으며, 마지막 5장에서 결론으로 마무리한다.

II. 관련 연구

Windows 환경에서의 다양한 PC 메신저에 대해 아티팩트 분석 및 데이터 복호화에 관한 연구가 수행되고 있다. Windows 8.1에서 Facebook과 Skype를 대상으로 로그인, 로그아웃, 연락처 목록과 대화 및 첨부파일과 관련된 아티팩트 분석결과가 제시되었다[3]. Windows 10에서 LINE 메신저를 사용한 후 저장된 데이터의 위치를 파악하고 주요 데이터를 식별한 연구결과가 제시된 바 있다[4]. 중국과 한국에서 인기 있는 메신저인 KakaoTalk, NateOn과 QQ를 대상으로 Windows 환경에서 생성되는 각각 데이터베이스 파일의 복호화 프로세스를 밝히고 복호화된 파일 내에서 주요 데이터를 식별한 연구결과가 제시된 바 있다[5].

또한, 네트워크 분석을 통해 아티팩트를 분석한 연구도 존재한다. WhatsApp의 프로토콜을 분석하여 암호화된 패킷에 대한 복호화 프로세스가 밝혀졌으며, 사용자 핸드폰 번호, 서버 IP, audio codec (Opus) 등의 데이터를 분석한 연구 결과가 제시되었다[6]. 카카오톡 채널 관리자, Purple과 TongTong 애플리케이션에 대한 데이터 복호화 방안과 네트워크 분석을 통해 로그인 시 전송되는 크리덴셜 데이터를 획득하는 방법이 제시된 바 있다[7]. 이외에도 구글 계정 애플리케이션을 대상으로 크리덴셜을 획득하여 다른 기기에서 정상 로그인을 할 수 있음을 보였으며, 이를 통해 개인정보 유출 가능성이 있음을 보였다[8].

본 논문에서는 일부 범죄 활동에 사용된 Wire 메신저를 대상으로 사용자 행위별 아티팩트 분석을 수행하였으며, 크리덴셜 획득을 통해 다른 기기에서 인증 없이 정상 로그인이 가능함을 보인다.

III. Wire 메신저 개요

2014년 12월에 출시된 Wire는 종단간 암호화를 지원하는 메신저다[9]. Android, iOS, Linux, macOS, Windows와 Web에서 사용 가능하다. 계정 생성 시, 메신저 내에서 친구 및 가족과 개인적인 메시지를 주고받는 Personal과 협업할 때 사용되는 Pro 중 선택하여 생성할 수 있다. 일대일 채팅, 그룹 채팅, 음성 통화, 영상 통화와 파일 공유 등의 기능을 제공하며, 전송한 메시지는 Fig. 1.과 같이 시간 간격을 설정하여 자동으로 삭제할 수 있다. 또한, 상대방이 지금 채팅이 가능한 상태인지 직관적으로 알아보기 위한 ping 기능도 제공한다.

전송한 메시지에 대해서는 Fig. 2.와 같이 좋아요, 메시지 수정, 메시지에 대한 답장, 삭제 기능을 제공한다. 메시지를 발신자의 채팅방에서만 삭제할 수도 있고, 발신자뿐만 아니라 수신자의 채팅방에서

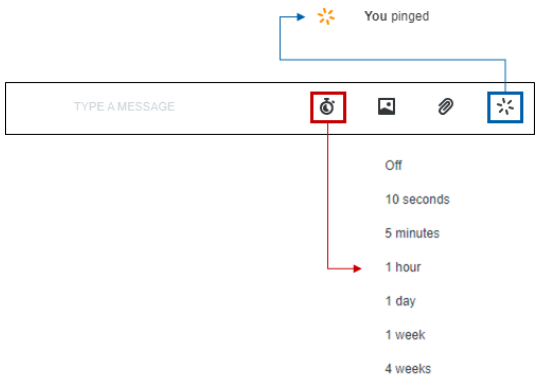

Fig. 1. Automatic message deletion time setting and ping function

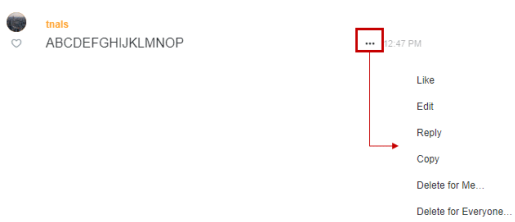

Fig. 2. Additional functions for sent message

삭제할 수도 있다.

장치마다 고유한 Device ID가 부여되며, 한 계정당 최대 7개의 장치를 등록할 수 있다. 새로운 환경에서 로그인 시 계정에 새 장치가 추가된다. 7개 이상 등록할 수 없으므로 이후 새로운 기기를 등록하고자 할 때는 기존에 등록된 장치 중 한 개를 삭제해야 한다. Wire의 설정 내에서 Fig. 3.과 같이 현재 장치의 Device ID와 등록된 장치 목록을 확인할 수 있다.

각 등록된 장치에는 Fig. 4.와 같이 활성화된 시간, 정보 장치마다 고유한 Key Fingerprint를 가지고 있으므로 이를 통해 장치 검증이 가능하고, session을 리셋할 수도 있다. 또한, 사용하지 않는 장치는 목록에서 삭제할 수 있다.

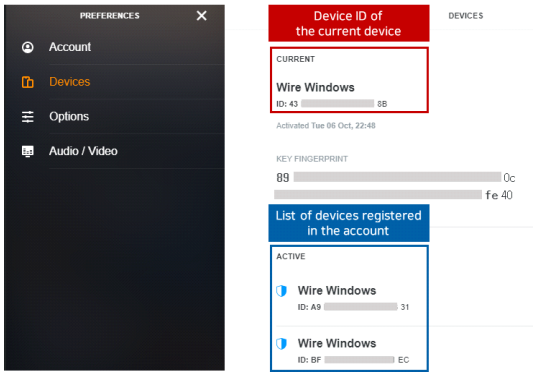

Fig. 3. Device ID of the current device and the list of devices registered under the account

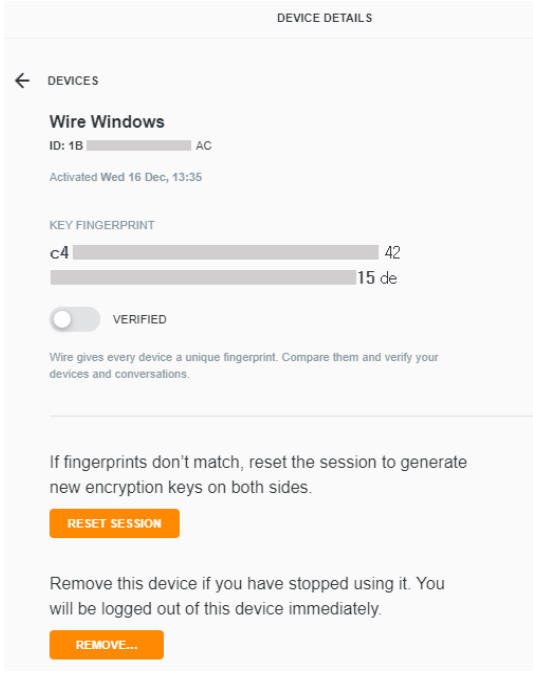

Fig. 4. Properties of registered devices

본 논문에서 사용한 연구 환경을 정리하면 Table 1.과 같다. Windows 10 환경에서 분석을 진행했으며, 2020년 12월을 기준으로 Windows에서의 최신 버전인 3.21.3932를 사용하여 분석했다.

Wire의 데이터 경로는 "%AppData%\Wire"이고 하위 구조는 Fig. 5.와 같다. Cookies 파일에는 크리덴셜 데이터가 저장되고, IndexedDB\https_app.wire.com_0.indexeddb.leveldb" 경로 내의 {0-9}{6}.log 파일에는 대화 내용이 기록된다. PC와 모바일 메신저의 데이터는 동기화되므로 저장된 대화 내역은 동일하다. 여러 개의 계정으로 로그인한 경우에는 Partitions 디렉터리 내에 계정별로 디렉터리가 생성되고 동일한 구조로 데이터가 저장된다.

본 논문의 분석 절차는 Fig. 6.과 같다. 준비 단계는 Wire 설치, 계정 생성과 기능 사용으로 나뉘고, 진행 단계는 데이터 식별, 수집 및 보존으로 나

Table 1. Analysis environment

Devices and Software	Name(Version)
Messenger	Wire(v3.21.3932)
PC	Windows 10 Pro
DB Browser	DB Browser for SQLite(v3.11.2)
Hex Viewer	HxD(v2.3.0.0)
Network Analysis	Fiddler Everywhere(v1.4.0)

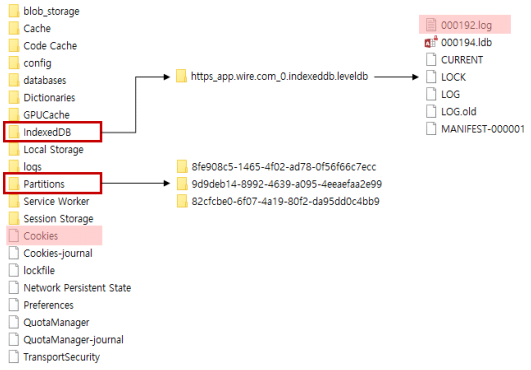

Fig. 5. Data structure of Wire

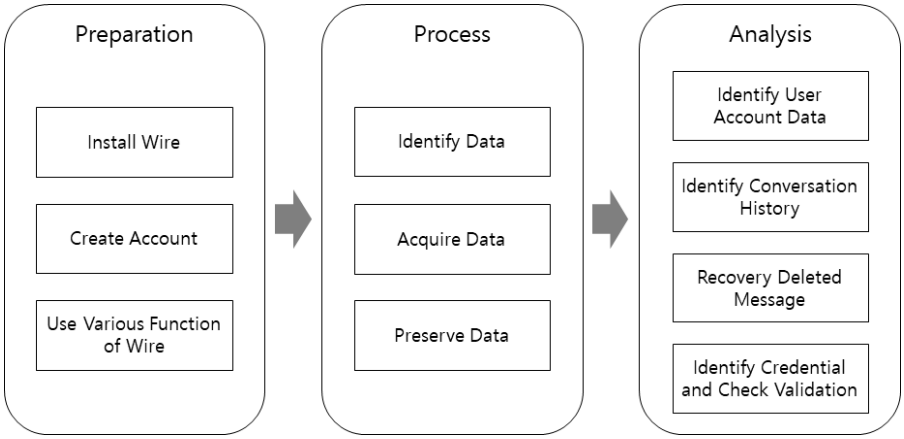

Fig. 6. Analysis procedure

뉜다. 마지막 분석 단계는 계정 정보, 대화 내역 및 크리덴셜과 같은 주요 아티팩트 식별, 삭제된 메시지 획득과 획득한 크리덴셜에 대한 유효성 확인으로 나뉜다.

IV. Wire 크리덴셜 획득 및 아티팩트 분석

본 장에서는 Wire의 크리덴셜 데이터를 획득하고 이를 통해 정상 로그인 상태로 애플리케이션 동작이 가능함을 보인다. Cookie 정보가 파일 형태로 남아 있고 이를 통해 로그인을 우회할 수 있으며, 네트워크 분석을 통해 사용자 ID와 패스워드를 획득할 수 있다.

4.1 Cookie

Wire의 크리덴셜 데이터는 Cookies 파일 내에 저장되어있으며, SQLite 데이터베이스 형태로 관리된다. Cookies 파일 데이터는 로그인 시마다 변경된다. 파일 내 저장된 데이터는 Fig. 7.과 같다.

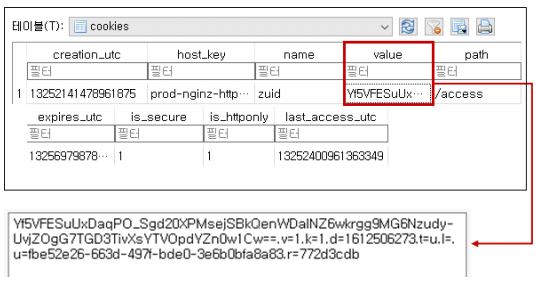

Fig. 7. Contents of cookies table

cookies 테이블 내의 creation_utc 컬럼은 cookie 생성 시각이 UTC로 저장되어있으며, value 컬럼은 cookie 데이터를 의미한다. 또한, expires_utc 컬럼은 만료 시각을 나타내며, last_access_utc 컬럼은 마지막으로 접근한 시각을 나타낸다. 두 컬럼에 저장된 시간 데이터는 Google Chrome 형식으로 저장된다.

Cookie 데이터의 구조는 Table 2.와 같다[10]. UUID (Universally Unique Identifier)와 Unix Time으로 표현된 토큰의 만료시간 등이 포함되어 있다. type은 a와 u로 두 종류이며, 각각 access token과 user token을 의미한다. 또한, tag에서 s는 session-based를 의미하고, 공백은 persistent를 의미한다. access data는 UUID와 임의의 8 bytes가 연접하여 저장되고, user data

Table 2. Data structure of cookie

Data	Content	Remark
String	signature	
v	version	
k	key-index	
d	timestamp	Unix Time
t	type	'a' or 'u'
l	tag	's' or ''
a	access data	UUID || c(8bytes)
u	user data	UUID || r(4bytes)

에는 UUID와 임의의 4 bytes 연접하여 저장된다.

4.2 사용자 로그인 정보

HTTP(S)에 대한 트래픽 캡쳐 및 모니터링을 지원해주는 웹 디버깅 프록시 도구인 Fiddler Everywhere를 통해 네트워크 패킷을 복호화하여 확인한 결과, Fig. 8.과 같이 로그인 시 ID로 사용한 username이나 email 정보와 패스워드를 전송하는 것을 확인할 수 있다[11]. 또한, 핸드폰 번호를 사용하여 로그인 시 사용자의 핸드폰 번호와 인증번호를 평문으로 전송하는 것을 확인할 수 있다.

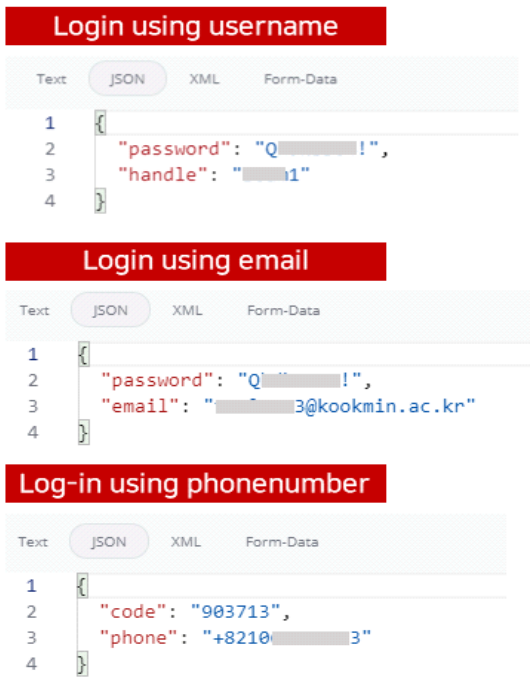

Fig. 8. Log-in information in a network packet

4.3 활용 가능 여부

Cookies 파일 획득 후, 계정에 분석 PC의 Device ID가 등록되어있으면 분석 PC에서 인증 과정 없이 로그인할 수 있다. Device ID가 등록되어있지 않은 상태에서는 "%AppData%\Wire" 경로의 하위 파일을 모두 획득해야 계정이 활성화된다. 그러나 로그인하려는 계정이 활성화되어 있거나 Log Out이 아닌 Fig. 9.와 같이 Quit Wire를 통해 종

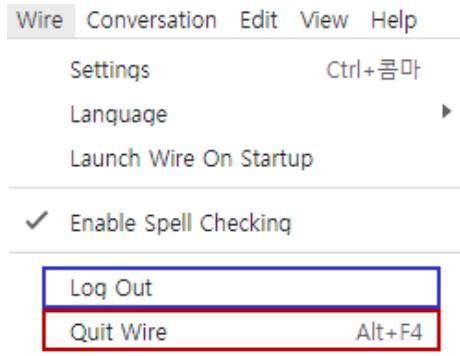

Fig. 9. Log Out and Quit functions of Wire

료된 상태에서만 가능하다.

여러 개의 계정이 로그인되어있는 경우에는 Partitions 디렉터리 내에서 계정마다 생성된 모든 Cookies 파일을 획득해야 한다. 획득한 Cookies 파일을 통해 여러 개의 계정 모두 정상 로그인 상태로 접근할 수 있다.

V. 사용자 행위 기반의 아티팩트 분석

Wire의 로그 파일은 leveldb 구조이며, 데이터 구조는 Fig. 10.과 같다[12]. 상위 7 bytes는 헤더 정보이며, 그중 4 bytes는 Checksum, 2 bytes는 데이터의 길이 그리고 1 byte는 데이터의 유형이 포함된다. 헤더 뒤에는 실제 데이터가 저장된다.

본 장에서는 사용자 행위별로 생성되는 데이터 유형을 확인하고, 각각의 유형별로 저장되는 아티팩트를 정리한다.

Fig. 10. Structure of leveldb

5.1 계정 정보

"%AppData%\Wire\logs"경로에 존재하는 electron.log 파일을 통해 계정 정보를 파악할 수 있다. 해당 기기에서 로그인했던 계정들에 대한 정보가 로그인할 때마다 생성된다. 로그인 시간은 UTC+0 형태이고, Fig. 11.과 같이 사용자의 이름과 사용자별로 부여된 고유 ID를 알 수 있다.

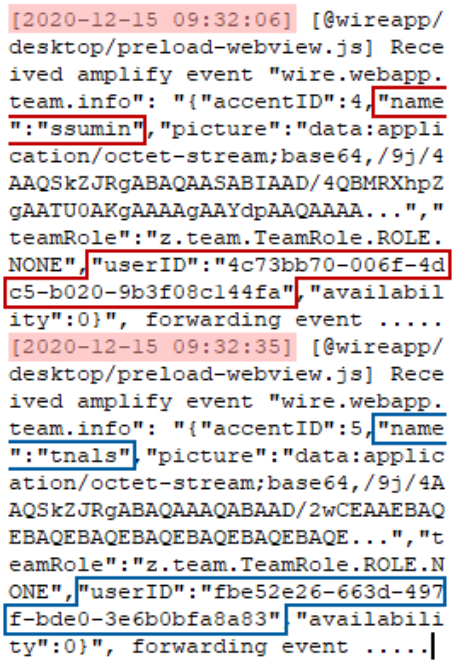

Fig. 11. Account information in electron.log

5.2 메신저 대화 내역

메시지 관련 데이터는 "%AppData%\Wire\IndexedDB\https_app.wire.com_0.indexeddb.leveldb" 경로 내의 [0-9]{6}.log 파일에 저장된다. 파일에 저장된 데이터는 Fig. 12.와 같다. 데이터 유형은 사용자 행위마다 다르게 지정되며, 유형마다 저장되는 내용도 다르다. 저장된 주요 아티팩트를 정리하면 다음과 같다.

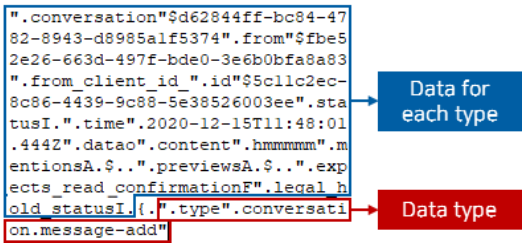

Fig. 12. Conversation data in [0-9]{6}.log file

5.2.1 conversation.message-add 유형

메시지를 수/발신한 경우와 특정 메시지에 대해 답장(reply)을 보낸 경우에 생성되는 유형이며, Table 3.과 같이 채팅방 ID, 발신인 ID, Device ID, 메시지 ID, 메시지 수/발신 시간과 메시지 내용 등이 포함된다. Device ID는 사용자가 수신한 메시지에 대해서만 저장되며, 발신 시간은 UTC+0 형태이다. 만약 채팅방 내에서 다른 사용자를 언급하면 언급한 사용자 이름이 포함되고, 메시지의 Like 기능을 사용한 경우에는 해당 사용자의 ID가 함께 저장된다. 또한, 메시지 자동삭제 기능을 사용한 경우 메시지 삭제 시간이 Unix Time 형태로 보여진다. 메시지를 수정하면 수정 시간과 함께 수정전의 메시지 ID가 기록된다. 이를 통해 메시지 변화 히스토리를 알 수 있다. 특정 메시지에 대해 답장을 보낸 경우에는 추가적으로 quote라는 문자열 뒤에 원본 메시지의 ID와 해당 메시지의 발신자 ID가 나타난다.

Table 3. Accompanying information of "conversation.message-add" type

Item	Content
conversation	Chatroom ID
from	Message sender ID
from_client_id	Device ID
id	Message ID
time	Message send time
content	Message content
mention	The name of a friend mentioned by a user in a message
reaction	User ID of a user who clicked a Like button
ephemeral_expires	Message deletion time (Unix Time)
edited_time	Message modification time
replacing_message_id	Message ID before modification
message_id	Original message ID
user_id	Sender ID of original message

5.2.2 conversation.asset-add 유형

해당 유형은 첨부파일을 수/발신 경우에 생성된다. 해당 유형에는 Table 4.와 같이 채팅방 ID, 발신인 ID, Device ID, 첨부파일 수/발신 시간과 첨부파일의 유형 및 파일명이 포함된다. 시간은 UTC+0 형태이며, 유형은 첨부파일의 확장자에 따라 다르며, image/jpeg, image/png, audio/wav, text/plain, text/xml, text/html, application/x-msdownload, application/x-zip-compressed, application/pdf 등이 있다. 또한, 첨부파일에 대해 Like 기능을 사용한 경우에는 해당 사용자의 ID가 함께 저장되고, 자동삭제 기능을 사용하면 첨부파일 삭제 시간이 Unix Time 형태로 기록된다.

Table 4. Accompanying information of "conversation.asset-add" type

Item	Content
conversation	Chatroom ID
from	Attachment sender ID
from_client_id	Device ID
id	Message ID
time	Attachment send time
content_type	Attachment type (image/jpeg, image/png, audio/wav, text/plain, text/xml,text/html, application/x-msdownload, application/x-zip-compressed, application/pdf etc)
name	Attachment name
reaction	User who used Like ID
ephemeral_expires	Message deletion time (Unix Time)

5.2.3 conversation.delete-everywhere 유형

수신한 메시지를 발신인이 모두에게서 수동 삭제한 경우 생성된다. 주요 데이터를 정리하면 Table 5.와 같다. 채팅방의 ID, 메시지 삭제 시간, 발신인 ID, 메시지 ID, 메시지 수/발신 시간이 기록된다. 이때, 삭제 시간과 수/발신 시간은 UTC+0 형태이다.

Table 5. Accompanying information of "conversation.delete-everywhere" type

Item	Content
conversation	Chatroom ID
deleted_time	Message deletion time
from	Message sender ID
id	Message ID
time	Message sent time

5.2.4 conversation.voice-channel-activate, conversation.voice-channel-deactivate 유형

통화 연결 여부에 따라 데이터의 유형이 다르다. conversation.voice-channel-activate 유형은 통화가 연결된 상태를 의미하며, conversation.voice-channel-deactivate 유형은 통화가 연결되지 않은 상태를 의미한다. 두 개의 유형은 Table 6.과 같이 채팅방 ID, 발신인의 ID, UTC+0 형태의 수/발신 시간이 저장된다.

Table 6. Accompanying information of "conversation.voice-channel-activate" and "conversation.voice-channel-deactivate" type

Item	Content
conversation	Chatroom ID
from	Caller ID
id	Message ID
time	Call time

5.2.5 conversation.one2one-creation 유형

일대일 채팅방을 생성한 경우 저장되고, Table 7.과 같이 채팅방 ID, 상대방의 ID와 채팅방을 생성한 사용자의 ID가 포함된다. 시간은 항상 1970-01-01 00:00:00으로 동일하게 나타난다.

Table 7. Accompanying information of "conversation.one2one-creation" type

Item	Content
conversation	Chatroom ID
userIds	Chat partner ID
from	User ID that created one2one chatroom
time	1970-01-01 00:00:00 (fixed value)

5.2.6 conversation.group-creation 유형

Wire에서 그룹이 생성한 경우에 저장되는 유형이다. 주요 데이터를 정리하면 Table 8.과 같다. 채팅방 ID, 그룹에 포함된 모든 사용자의 ID, 그룹을 생성한 사용자의 ID, 그룹명, 그룹을 생성한 시간이 포함된다. 그룹 생성시간은 UTC+0 형식이다.

Table 8. Accompanying information of "conversation.group-creation" type

Item	Content
conversation	Chatroom ID
name	Group name
userIds	All user ID in the group
from	User ID that created group
time	Group creation time

5.2.7 conversation.rename 유형

그룹채팅방의 이름을 변경했을 때 생성되는 유형이다. 주요 데이터를 정리하면 Table 9.와 같다. 채팅방 ID, 이름을 변경한 시각, 변경된 이름, 변경한 사용자 ID가 포함된다.

Table 9. Accompanying information of "conversation.rename" type

Item	Content
conversation	Chatroom ID
time	Group chatroom name change time
name	Changed group chatroom name
from	User ID that changed the group chatroom name

5.2.8 conversation.knock 유형

ping 기능을 사용했을 때 생성되는 유형이며, 주요 데이터를 정리하면 Table 10.과 같다. ping을 전송한 채팅방의 ID, 전송한 사용자의 ID, Device ID, UTC+0 형태의 전송시간이 포함된다. Device ID는 ping을 수신한 경우에만 저장되고 발신한 경우에는 저장되지 않는다.

Table 10. Accompanying information of "conversation.knock" type

Item	Content
conversation	Chatroom ID
from	User ID sent ping
from_client_id	Device ID
id	Message ID
time	Ping sent time

5.2.9 verification_state 유형

로그인한 계정이나 생성한 그룹에 대한 정보가 저장되는 유형이다. 주요 데이터를 정리하면 Table 11.과 같다. 마지막으로 활동한 시간, 마지막으로 읽은 시간과 사용자명 또는 그룹명이 포함된다. 또한, 채팅방에 포함된 사용자의 ID가 기록된다.

Table 11. Accompanying information of "verification_state" type

Item	Content
last_event_timestamp	Last active time
last_read_timestamp	Last time that user read message
muted_timestamp	Chatroom muted time
name	Chatroom name (user name or group name)
others	All user ID in the chatroom

5.2.10 conversation.unable-to-decrypt 유형

오류 발생한 경우에 생성되는 유형이며, 주요 데이터를 정리하면 Table 12.와 같다. 채팅방 ID, 오류 발생 이유, 오류 코드와 발생 시간 등의 정보를 포함한다.

Table 12. Accompanying information of “conversation.unable-to-decrypt” type

Item	Content
conversation	Chatroom ID
error	The reason for error
error_code	Error code
from	Sender ID
id	Message ID
time	Error occurrence time

5.3 대화 내역 삭제

전송한 메시지를 정한 시간에 맞춰 자동으로 삭제하거나 수동으로 삭제하는 기능을 사용하면 Fig. 13.과 같이 채팅방에서는 삭제되는 것을 확인할 수 있다.

수/발신한 메시지를 자동 삭제한 경우에는 log 파일에 Fig. 14.와 같이 삭제 시간인 ephemeral_expires가 함께 기록되어 있으므로 삭제된 메시지 임을 알 수 있다. 또한, 삭제된 메시지도 남아 있으므로 획득할 수 있다.

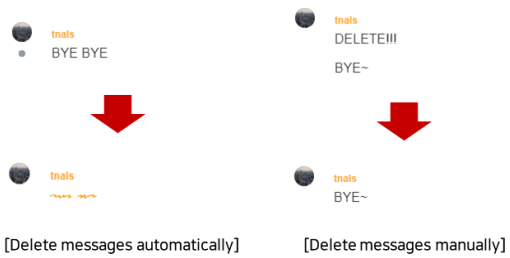

Fig. 13. Information on directly deleted data

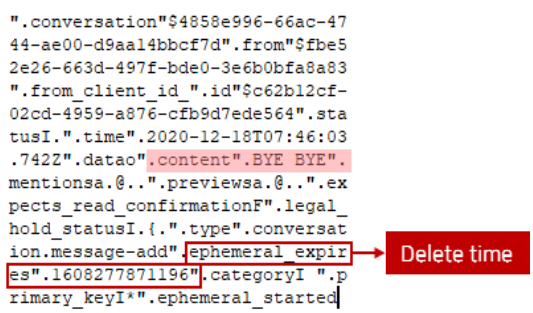

Fig. 14. Log file contents for automatically deleted message

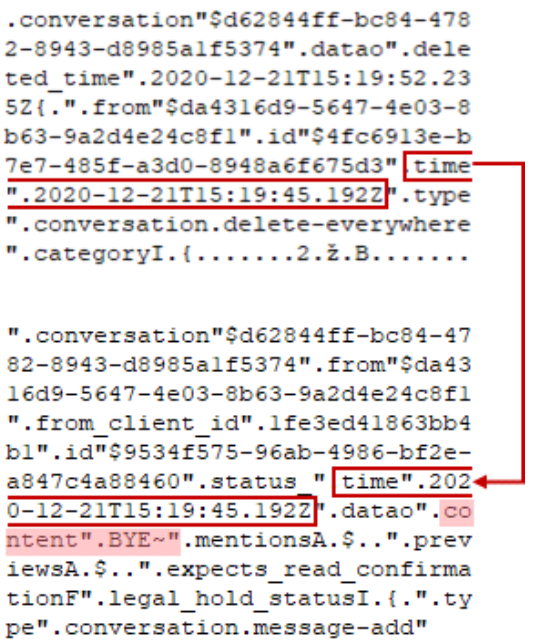

Fig. 15. Information on automatically deleted data

수신한 메시지를 발신인이 모두에게서 수동 삭제한 경우에는 수신인 log 파일에 “conversation.delete-everywhere” 유형으로 기록된다. 유형에 기록된 메시지 발신 시간을 확인한 후, 해당 시간을 검색하여 Fig. 15.와 같이 삭제된 메시지를 획득할 수 있다.

VI. 결 론

본 논문에서는 로컬에 저장된 Cookies 파일에서 획득한 크리덴셜 정보를 통해 별도의 PC에 정상 로그인 상태로 접근이 가능함을 보였으며, 네트워크 분석을 통해 사용자의 로그인 정보를 획득할 수 있었다. 또한, 사용자 행위를 기반으로 생성되는 아티팩트인 계정 정보와 대화 내역 정보에서 주요 데이터를 선별하고 분류하였다. 마지막으로 메시지를 자동/수동 삭제해도 채팅방에선 남아 있지 않지만, 로그 파일을 통해 획득할 수 있음을 확인하였다. 본 논문의 결과를 통해 포렌식적으로 유용한 데이터를 효율적으로 수집할 수 있도록 도움 줄 것으로 기대한다.

References

[1] "Aju Business Daily", https://www.ajunews.com/view/20201014141141075

[2] "Boannews", https://www.boannews.com/media/view.asp?idx=87562&page=1&kind=1

[3] Teing Yee Yang, Ali Dehghantanha, Kim-Kwang Raymond Choo and Zaiton Muda, "Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies," *PLoS one*, 11(3), e0150300, Mar. 2016

[4] Ming Sang Chang and Chih Yen Chang, "LINE Messenger Forensics on Windows 10," *Journal of Computers*, 30(1), pp. 114-125, Feb. 2019

[5] Jusop Choi, Jaegwan Yu. Sangwon Hyun, and Hyoungshick Kim, "Digital forensic analysis of encrypted database files in instant messaging applications on Windows operating systems: Case study with KakaoTalk, NateOn and QQ messenger," *Digital Investigation*, 28, pp. 550-559, Apr. 2019.

[6] Karpisek, Filip, Ibrahim Baggili, and Frank Breitinge, "WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages," *Digital Investigation*, 15, pp. 110-118, 2015

[7] Sehoon Lee, Myungseo Park, Giyoon Kim, Uk Hur, and Jongsung Kim, "A Study on The Decryption of Encrypted SNS Application Data : KakaoTalk Channel, Purple, TongTong" *Journal of Digital Forensics*, 14(1), pp. 87-96, Mar. 2020.

[8] Jong-Won Choi and Jeong-Hyun Yi, "Analysis on Personal Information Leakage of Google Account App on Android," *Journal of Digital Forensics*, 8(2), pp. 65-81, Dec. 2014

[9] "Wire", https://wire.com/en/

[10] "Wire Security Whitepaper", https://wire-docs.wire.com/download/Wire+Security+Whitepaper.pdf

[11] "Fiddler", https://www.telerik.com/fiddler

[12] "Leveldb GitHub", https://github.com/google/leveldb

〈저자소개〉

신 수 민 (Sumin Shin) 학생회원
2020년 2월: 국민대학교 정보보안암호수학과 졸업
2020년 3월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호

김 소 람 (Soram Kim) 학생회원
2016년 2월: 국민대학교 수학과 졸업
2018년 2월: 국민대학교 금융정보보안학과 석사
2018년 3월~현재: 국민대학교 금융정보보안학과 박사과정
〈관심분야〉 디지털 포렌식, 정보보호

윤 병 철 (Byungchul Youn) 학생회원
2019년 2월: 국민대학교 수학과 졸업
2019년 9월~현재: 국민대학교 금융정보보안학과 석사과정
〈관심분야〉 디지털 포렌식, 암호학

김 종 성 (Jongsung Kim) 종신회원
2000년 8월/2002년 8월: 고려대학교 수학 학사/이학석사
2006년 11월: K.U.Leuven, ESAT/SCD-COSIC 정보보호 공학박사
2007년 2월: 고려대학교 정보보호대학원 공학박사
2007년 3월~2009년 8월: 고려대학교 정보보호기술연구센터 연구교수
2009년 9월~2013년 2월: 경남대학교 e-비즈니스학과 조교수
2013년 3월~2017년 2월: 국민대학교 수학과 부교수
2013년 3월~2020년 8월: 국민대학교 일반대학원 금융정보보안학과 부교수
2017년 3월~2020년 8월: 국민대학교 정보보안암호수학과 부교수
2020년 9월~현재: 국민대학교 정보보안암호수학과/일반대학원 금융정보보안학과 교수
〈관심분야〉 정보보호, 암호 알고리즘, 디지털 포렌식