사용자 행위 중심의 협업 툴 'JANDI' Artifact 분석

이광호*, 나소진**, 정예진***, 이은빛****, 문현지*****, 이경현******,

*부경대학교(학부생), **충남대학교(학부생), ***경기대학교(학부생),
****순천향대학교(학부생), *****서울지방병무청, ******부경대학교(교수)

Artifact analysis of User Behavior-Centric Collaboration tool 'JANDI'

Kwang-Ho Lee*(Ungraduate student), So-Jin Na**(Ungraduate student), Ye-Jin Jeong***(Ungraduate student), Eun-Bit Lee****(Ungraduate student), Hyun-Ji Moon*****, Kyung-Hyune Rhee******(Professor)

*Pukyong National University, **Chungnam National University, ***Kyonggi University, ****Soonchunhyang University, *****Seoul Regional Military Manpower Administration, ******Pukyong National University

요 약

최근 협업 툴 아티팩트를 분석하여 사용자 간 대화 내용을 기반으로 기업 범죄 사실을 입증한 사례가 있었다. 이에 포렌식 관점에서 사용자 행위 중심의 협업 툴 아티팩트 분석에 대한 필요성이 대두되었다. 본 논문에서는 Windows 환경에서 국내 협업 툴 JANDI를 대상으로 Chromium 기반의 웹 실행 환경과 Electron Framework 기반의 애플리케이션 실행 환경의 아티팩트를 분석하였다. 이를 통해 프로필 이미지 파일과 전송한 이미지의 썸네일을 확인할 수 있었다. 또한, 사용자의 메시지 삭제 행위를 확인하고 전송한 메시지를 식별할 수 있는 도구를 각각 개발하여 협업 툴 아티팩트 분석 시나리오에 적용함으로써 디지털 포렌식 수사에서 증거로 활용할 수 있는 방안을 제시하였다.

I. 서론

협업 툴은 사내 영업이익 혹은 기술적 기밀 정보를 포함한 다양한 정보를 담고 있다. 사례로 키로그 프로그램을 통해 협업 툴 '네이트온' 아이디를 획득한 후, 사용자 계정 접속 및 대화 내용을 다운로드를 받은 행위에 대해 피고인의 전자기록탐지죄를 인정한 판례[1]가 존재한다. 이처럼 협업 툴이 해커들의 공격 대상 혹은 부정비리로 인한 내부감사의 분석 대상이 될 수 있기 때문에 협업 툴에 대한 디지털 포렌식 관점의 아티팩트 분석 활용 방안 연구가 필요하다.

JANDI는 2015년 국내에서 출시된 업무용 협업 툴이며 개인, 단체 대화방뿐만 아니라 주제별 대화방인 토픽, 클라우드를 통한 파일 관리 등 협업에 필수적인 기능을 갖추고 있다.

본 연구에서는 협업 툴 JANDI의 Chromium 기반의 웹 실행 환경과 Electron Framework 기반의 애플리케이션 실행 환경을 기준으로 변경된 아티팩트에 대해 사용자 행위 중심으로 분석하는 방안을 제시하였다.

논문의 구성은 다음과 같다. 2장에서는 JANDI와 관련된 선행 연구를 설명한다. 3장에서는 메시지 전송 및 삭제와 같은 사용자의 행위 식별, 전송한 메시지와 파일 등의 사용자 행위 분석 방법을 설명한다. 4장에서는 분석한 아티팩트를 활용한 시나리오 사건 분석, 5장에서 결론 및 향후 연구로 마무리한다.

II. 관련 연구

JANDI는 협업 툴 Slack등과 마찬가지로 다양한 기기에서의 접근을 위한 크로스 플랫폼 형태로 제작되었다는 특징을 가지고 있다. 또한

JANDI와 같은 Electron Framework 형태의 애플리케이션은 Chromium 기반의 웹 브라우저와 동일한 형태로 Cache, Local Storage, IndexedDB, Session Storage, Cookies에 데이터를 저장한다.[2] 기존 협업 툴과 관련된 연구에서 김성수 등[3]은 Windows 환경에서 협업 툴 JANDI, Slack, Microsoft Teams를 중심으로 Electron App의 메시지 획득 방법을 제시하였다. 그러나 JANDI의 경우 Windows 환경에서 메시지 삭제 이후 복구가 불가능하고, 삭제된 메시지와 관련된 흔적을 찾지 못했다고 기술하였다. 신수민 등[4]은 Windows 환경에서 Wire 메신저를 대상으로 LevelDB의 log파일을 분석하여 대화 내역을 확인하고, Web Proxy를 통해 사용자 로그인 정보를 확인하였으며, 행위별로 생성되는 로컬 데이터의 분석 결과를 제시하였다. 조민지 등[5]은 Windows환경, macOS, Web Browser 환경에서 Wire 메신저의 LevelDB에 저장된 아티팩트를 종합적으로 분석하여 사용자의 행위를 특정하는 방안을 연구하고, 아티팩트 분석 도구를 개발하였다. 김한결 등[6]은 Windows 환경에서의 협업 툴의 기능과 사용자 권한과 같은 협업 툴의 특성을 고려하여 데이터를 수집하였다. 그리고 Chrome 환경에서의 캐시 데이터에서 송/수신 이미지 파일을 획득하였다.

박귀은 등[7]은 iOS 환경에서 Naver WORKS와 JANDI를 분석하여 주요 아티팩트를 식별하고 삭제된 데이터에 대한 식별과 복구 방안을 제시하고, Electron Framework를 이용한 PC 환경의 메신저에 대해서 LevelDB를 이용해 사용자의 행위를 식별하는 관련 연구를 수행하였다. 홍리나 등[8]은 Android 환경에서 Channel Talk를 중점으로 JANDI, Slack, Naver WORKS의 아티팩트 분석 후 비교 분석을 진행하며, JANDI 사용자의 계정 정보 저장 위치 등을 기술하였다. 신수민 등[9]은 Android 환경에서 Naver WORKS와 JANDI를 대상으로 아티팩트 분석을 진행하고 유의미한 데이터를 식별하여 채팅방 재구성 및 삭제된 메시지 복구 방안을 제시하였지만, Windows 환경에서의 삭제된 메시지 복구 방안에 관한 연구는 진행되지 않았다.

이처럼 Windows, iOS, Android 등 여러 환경에서 JANDI 아티팩트를 분석하고 삭제된 메시지 및 사용자 행위 식별에 관한 연구가 진행되었다. 그러나 Windows 환경에서의 변화된 데이터 저장 방식과 그에 따른 삭제된 메시지와 관련된 추가 연구의 필요성을 느꼈다. 본 연구가 차후 JANDI와 같은 Electron Framework 기반의 협업 툴 수사에 도움이 되기를 바란다.

III. 사용자 행위별 아티팩트 분석

분석 대상은 행위 이후 로컬 PC에 남아있는 캐시 데이터, 이벤트 로그를 대상으로 한다. Windows 환경에서 분석을 위해 FTK Imager로 이미지 파일을 추출하였으며, 데이터를 확인하기 위해 Chrome Cache View와 Hex Editor를 사용하였다. 또한, 3.2.1장에서 서술할 'utf-16-le' 인코딩이 적용된 메시지의 디코딩을 위해 LevelDB에서 메시지를 추출 후 디코딩 과정을 자동화한 스크립트[10]를 제작하였다. 연구에 사용된 소프트웨어를 Table 1.과 같이 정리할 수 있다.

Table 1. Experiment Setup

Software	Name	Version
Collaboration Tool	JANDI (Electron App)	1.7.6
	JANDI (Chrome)	24.3.1
Image Mounting Tool	FTK Imager	4.7.1.2
Hex Viewer	HxD(hex editor)	2.5.0.0
Browser Cache Inspection Tool	Chrome Cache View	2.46

3.1 사용자 행위 식별

JANDI에서는 메시지 전송 및 삭제, 새 창 띄우기 등의 일부 특정한 행위에 대한 event log를 Cache에서 확인할 수 있다. 사용자 행위 이후 Cache에 생성된 데이터는 “https://track.jandi. com/log/web?callback=jQuery{FunctionNa

me}&footprint={log}={UnixTime}” 형식으로 작성된다.

footprint 이후 작성된 log를 Base64로 디코딩했을 때, event code를 포함한 변수들을 확인할 수 있다. 캐시 데이터에서 log를 추출하여 디코딩하는 과정을 자동화한 Python 기반 log 추출 도구[11]를 개발하여 데이터를 분석하였다. 그 결과 event code에 따라서 구성하는 변수가 상이함을 확인하였으며, 이 외 일정하게 발생하는 변수를 Table 2.과 같이 정리하였다.

Table 2. Event log variables

Column	Value	Remark
ev	Event Log	e44: send message e46: delete message
t	Transection ID	-
s	Session ID	-
a	JWT sub	-
m	Writer ID	-
pl	Execuiton Environment	winapp, web
pr	Properties	-
time	Recording time	-
version	Program version	-

Table 2.의 사용자 행위에 대한 이벤트 코드와 log 생성 시간을 의미하는 변수 등을 통해 log를 분석하여 메시지 전송·삭제와 같은 사용자 행위와 그 시간에 대해 식별할 수 있음을 확인하였다.

3.2 사용자 행위 분석

JANDI에서 분석 가능한 사용자의 행위에는 전송한 메시지 내역, 전송한 이미지 파일의 썸네일, 설정한 프로필 이미지가 존재한다.

3.2.1 메시지

JANDI는 개인과 단체 대화방, 주제별 대화방인 토픽의 방법으로 메신저 기능을 수행한다. LevelDB 분석을 통해 해당 대화 내역을 확인할 수 있다. LevelDB 파일에서는 부분적으로 식별 가능한 압축되지 않은 데이터가 존재하는데, 본 논문에서는 이를 분석하여 로컬 PC에 저장된 메시지 데이터를 획득한다. JANDI 실행 환경에 따라 Chrome 사용 시 "AppData\Local\Google\Chrome\User Data\Default\Local Storage\leveldb", Electron App 사용 시에는 "AppData\Roaming\JANDI\Local Storage\leveldb" 경로 내의 log 파일 및 ldb 파일에 저장된다.

각각 경로 내 파일에서 메시지 입력창에 입력된 텍스트를 확인할 수 있다. 한글의 경우 영문과 다르게 영어로 작성된 메시지 키로그는 육안으로 확인할 수 있었지만, 한글로 작성된 경우 불가능했다. 한글 메시지 키로그는 역공학 기법을 통해 ‘utf-16-le’ 인코딩 방식으로 LevelDB의 log 파일에 기록됨을 확인하였다. LevelDB에 기록된 헥사값을 본 논문에서는 자체 개발한 코드[10]를 통해 한글로 변환이 가능하다.

3.2.2 파일

전송된 이미지 파일 썸네일을 Cache에서 확인할 수 있다. 캐시 데이터는 JANDI 실행 환경에 따라 Chrome 사용 시 "AppData\Local\Google\Chrome\User Data\Default\Cache", Electron App 사용 시에는 "AppData\Roaming\JANDI\Cache" 경로에서 확인할 수 있다.

썸네일의 캐시 데이터는“https://jandi-box.com/files-thumb/{TeamId}/{FileName}?size=640”의 구조를 가지는데, 해당 URL로 접속 시 검증을 거치지 않고 전송한 이미지 파일을 획득할 수 있다. 아울러 FileName의 앞 10byte는 파일이 전송된 시간을 Unix time으로 나타낸다. 이를 통해 해당 이미지 파일이 전송된 시간을 확인할 수 있다. 이는 채팅방에 남아있는 이미지뿐만 아니라 사용자가 발송 후 삭제한 이미지 파일에도 적용할 수 있다.

3.2.3 프로필 이미지

또한 JANDI 사용자가 설정한 프로필 이미지 파일을 Cache에서 확인할 수 있다. 캐시 데이터는 “jandi-box.com/files-profile/”의 구조를 가진다.

IV. 아티팩트 활용 시나리오

4.1. 사건 개요

국내 운송 업체 A사는 JANDI를 A사의 공식 협업 툴로 사용 중이다. A사 사원 K 씨는 자사 서비스 이용자 중 유명인의 개인정보를 무단으로 열람한 혐의가 있다. 또한 K 씨는 사내 메신저 JANDI를 통해 다른 직원에게 무단으로 열람한 내용을 텍스트 메시지와 이미지 파일로 캡처하여 전송한 것으로 의심된다.

4.2. 사건 분석

혐의자의 동의를 얻어 압수한 PC에 로그인 후, LevelDB 데이터와 캐시 데이터를 획득한다. 먼저 혐의자가 전송한 메시지 데이터 확인을 위해 본 논문에서 개발한 스크립트[10]를 활용하여 LevelDB에 기록된 메시지를 수집한다. 또한, 혐의자가 전송한 이미지 파일 획득을 위해 Chrome Cache View를 사용하여 캐시 데이터를 분석한다. 데이터 중 ‘jandi-box.com/files-thumb’ 문자를 포함한 캐시 데이터를 확인 후 URL 형태로 저장된 데이터에 직접 접근하여 혐의자가 발송한 이미지 파일을 획득한다.

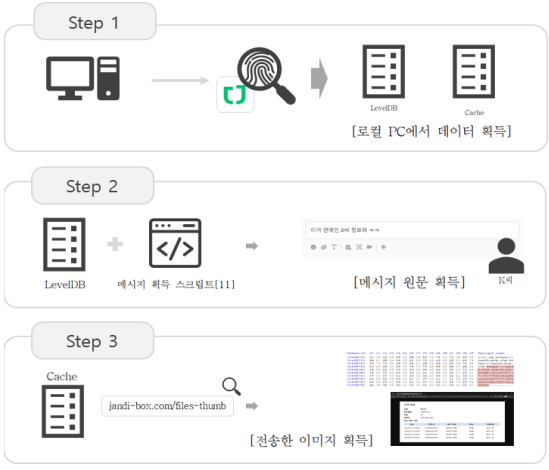

Fig. 1. The process of analyzing evidence

본 논문에서 제시한 방법을 통해 혐의자가 협업 툴 JANDI에서 발송한 메시지와 이미지 파일을 획득할 수 있고, 이를 근거로 혐의자의 범죄를 입증할 수 있음을 확인하였다.

V. 결론

JANDI에 대한 선행 연구에서는 Windows 환경에서 Cache나 Database에 접근하여 전송한 메시지와 사용자 정보를 확인할 수 있었으나, 현재는 동일한 방법으로 정보를 획득이 불가능하다. 또한, 삭제된 메시지 확인이 불가능했지만, 본 논문에서는 삭제된 메시지에 대해 일부 식별이 가능했다.

‘마약류 관리에 관한 법률 위반’으로 피고인들에게 징역을 선고한 사례 중 범죄 과정에서 피고인들이 마약 수입 및 투약 관련 정보를 ‘위챗’ 협업 툴로 공유한 사례가 있다.[12] 해당 사건의 수사 과정에서 협업 툴 아티팩트를 분석함으로써 사용자 간 대화 내용을 기반으로 피고인들의 범죄 사실을 입증할 수 있었다. 이를 통해 사용자 행위 중심의 협업 툴 아티팩트 분석이 디지털 포렌식 관점에서 의의가 있음을 확인하였다. 본 논문에서는 Cache에서 사용자의 메시지 삭제 행위를 확인할 수 있는 도구를 개발하고, 프로필 이미지 파일과 전송한 이미지 파일의 썸네일을 확인할 수 있었다. 또한 LevelDB의 log 파일에서 전송한 메시지를 식별할 수 있는 도구를 개발하여 사용자의 대화 내역을 식별할 수 있었다.

4장에서 본 연구에서 얻은 JANDI 아티팩트를 시나리오 분석에 적용함으로써 디지털 포렌식 수사에서 증거로 활용될 수 있는 데이터임을 확인할 수 있었다. Slack, Discord 등 다수의 협업 툴이 JANDI와 동일한 Electron Framework를 기반으로 하고 있다는 점에서 이 연구가 협업 툴 메신저 분석 시 디지털 포렌식 관점에서 기여하기를 기대한다.

그러나 일부 메시지의 경우 Snappy 압축 라이브러리를 통해 log 파일이 ldb 파일로 압축된 후에는 육안으로 식별하기가 어렵다는 한계가 존재한다. ldb 파일도 log 파일과 동일하게 메

시지를 식별할 수 있는 방법은 향후 연구로 남긴다.

[참고문헌]

[1] Supreme Court of Korea, Mar. 31, 2022, 2021 Criminal Appeal 8900

[2] web.dev, “Storage for theWeb,” https://web.dev/storage-for-the-web, accessedSep.2021

[3] Sungsoo Kim and Sungjin Lee, "A Study on Message Acquisition from Electron Apps:Focused on Collaboration Tools such as Jandi, Slack, and Microsoft Teams", Journal of The Korea Institute of Information Security & Cryptology, 32(1), pp.11-23, 2022

[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, 2021

[5] Minji Cho, Byeongchan Jeong, Jungheum Park and Sangjin Lee, "A Research on User Data Caching Mechanism of Wire Messenger in PC", Journal of Digital Forensics, 16(2), pp.49-59, 2022

[6] Han-gyeol Kim, Dabin We, Insoo Lee, Naeun Kim, and Myungseo Park, “Study on Methods for Collecting Collaboration Tool Data in Windows Environment,” Contemporary Review of Forensic Science, No. 9, pp. 3-3, May 2024.

[7] Gwuieun Park, Minjeong Lee, Soojin Kang, Soram Kim and Jongsung Kim, "A Study on Artifacts Analysis and Credential Utilization Method of Collaboration Tools in iOS", Journal of Digital Forensics, 17(2), pp.14-32, 2023

[8] Hong Rina, Son Tae-sik, "Analysis of messenger-type collaboration tool application artifacts - Focusing on ChannelTalk", Journal of Digital Forensics, 18(1), pp.79-96, 2024

[9] 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, 2021

[10] https://github.com/hoihosick/whs-project-jandi/blob/main/decode.py

[11] https://github.com/hoihosick/whs-project-jandi/blob/main/track_parser.py

[12] Uijeongbu District Court Goyang Branch, Dec. 29, 2021, 2021 Criminal Panel 215, 274 (consolidated)