

안드로이드 기반 클라우드 스토리지 앱 크리덴셜 활용 및 아티팩트 분석

최 용 철*, 김 기 윤*, 김 중 성**,*
국민대학교 금융정보보안학과(대학원생)*, (교수)**,
국민대학교 정보보안암호수학과 (교수)**

Android-based Cloud Storage App Credential Utilization and Artifact Analysis

Yongcheol Choi*, Giyeon Kim*, Jongsung Kim**

Dept. of Financial Information Security, Kookmin University (Graduate Student)*, (Professor)**

Dept. of Information Security, Cryptology and Mathematics, Kookmin University (Professor)**

요 약

불법촬영 및 성착취물과 같은 디지털성범죄에서 용의자의 스마트폰은 주요 증거가 될 수 있다. 용의자가 모바일 클라우드 스토리지를 사용하는 경우 데이터가 저장장치에 저장되지 않아 수사에 지장을 준다. 실제로 모바일 클라우드 스토리지를 이용하여 수사를 회피 및 방해한 사례가 존재한다. 모바일 클라우드 스토리지는 각 앱마다 아티팩트가 다르므로 데이터 수집에 어려움이 있다. 따라서 기존에 연구되지 않은 모바일 클라우드 스토리지 앱의 사전 연구는 지속적으로 요구된다. 널리 사용되에도 불구하고 분석된 바 없는 3종 TeraBox, DeGoo, Yandex.Disk에 대해 분석하였다. 각 앱에 대해 크리덴셜을 획득하여 재사용하는 방안을 설명하였으며 저장된 사용자 데이터를 분석하였고, 가상 시나리오를 작성하여 수사관점에서 획득한 데이터를 활용하는 방안을 제시하였다.

주제어 : 디지털 포렌식, 아티팩트 분석, 클라우드 앱

ABSTRACT

In digital sexual abuses such as illegal filming and sexual molestation, a suspect's smartphone can be the main evidence. If the suspect uses mobile cloud storage, the data is not stored in the local device, which interferes with the investigation. In fact, there are cases of avoiding and interfering with investigations using mobile cloud storage. Mobile cloud storage has different artifacts for each app, making it difficult to collect data. Therefore, prior research on mobile cloud storage apps that have not been previously studied is continuously required. This paper analyzed three types of TeraBox, DeGoo, Yandex.Disk with no analyzed results despite of widely use. For each app, a method of acquiring and reusing credentials was explained, stored user data was analyzed, and a virtual scenario was created to suggest a method of utilizing the data obtained from an investigation perspective.

Key Words : Digital Forensics, Artifact Analysis, Cloud App

1. 서 론

코로나바이러스의 장기간 지속에 따른 정책으로 인해 재택근무와 원격업무가 증가함에 따라 클라우드 서비스의 사용량이 증가하고 있다[1]. 하지만 사용량 증가에 따라 클라우드 서비스를 이용한 범죄 역시 증가하고 있다[2]. 클라우드 서비스의 한 종류인 클라우드 스토리지는 사용자가 업로드한 파일을 링크를 생성하여 링크 소

※ 본 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2021-0-00540, GPU/ASIC 기반 암호알고리즘 고속화 설계 및 구현 기술개발)

• Received 03 March 2022, Revised 04 March 2022, Accepted 31 March 2022

• 제1저자(First Author) : YongCheol Choi (Email : chldydcjf@kookmin.ac.kr)

• 교신저자(Corresponding Author) : Jongsung Kim (Email : jskim@kookmin.ac.kr)

유자들에게 공유할수 있다. 따라서 클라우드 서비스를 중간 유통망으로 사용하여 범죄를 저지르는 사례들이 존재한다[3]. 특히 클라우드 스토리지는 불법 촬영물이나 성착취물과 같은 디지털 성범죄 자료를 저장 및 공유에 악용된다. 이 중에서 모바일 클라우드 스토리지에 관한 사례는 여럿 존재한다. 국내에서는 불법 촬영물을 클라우드 스토리지에 업로드하여 경찰 수사를 회피하려 한 사례가 존재한다[4]. 해당 사례는 여성의 신체 부위를 촬영한 사진이나 영상을 자동으로 클라우드에 업로드하고 원본 파일을 삭제하는 형식으로 수사망을 회피하려 한 사례이다. 또한 해당 계정의 비밀번호를 잊었다고 하여 수사를 방해한 사례도 존재한다[5]. 불법 촬영의 정황상 근거는 있으나, 촬영물을 찾지 못하여 수사에 난항을 겪었다. 클라우드 스토리지를 악용한 사례가 늘어나고 다양한 클라우드 스토리지가 생성됨에 따라 원활한 수사를 위해 사전연구의 수요가 늘고 있다. 본 논문에서는 이용자 수가 많지만 연구결과가 없는 모바일 클라우드 스토리지 앱 3종 TeraBox, DeGoo, Yandex.Disk를 분석했다. 본 논문에서는 2장에서는 관련 연구 및 분석환경을 서술하고, 각 3장, 4장, 5장에서는 앱 3종에 대한 아티팩트 분석 및 크리덴셜 활용을 제시하며, 6장에서 결론으로 마무리 짓는다.

II. 관련 연구 및 분석환경

2.1 관련 연구

클라우드 스토리지 앱에 대한 연구는 다양하게 진행되고 있다. Windows 10 환경에서 Box 클라우드 아티팩트를 분석한 연구[6]에서는 Windows 10 PC 기반에서 사용할 수 있는 Box 클라우드 서비스의 아티팩트를 분석하였으며, 해당 아티팩트를 토대로 Box 클라우드와 관련된 사건 시나리오를 분석하였다. 구글 클라우드 데이터의 수사 활용방안을 연구가 진행된 바 있다[7]. 해당 연구는 구글 클라우드의 스마트폰 사용자를 중심으로 분석하였다. 구글 클라우드 아티팩트 분석과 데이터 획득 방법, 획득한 데이터를 분석하여 시각화를 진행하였으며 분석된 아티팩트의 수사적 의미 및 사건 유형별로 아티팩트를 정리하였다. 클라우드 아티팩트 자동 수집 및 분석 시스템을 개발한 연구[8]에서는 업데이트가 빈번하게 이루어지는 클라우드 서비스에 대해 변화될 수 있는 아티팩트 탐지를 자동화하는 연구를 진행하였다. 구축된 시스템은 설치되어 있는 클라우드 서비스가 현재 Database 내에 있는 버전과 일치한지 검증하고 업데이트되었을 경우 관리자에게 메시지를 전송한다. 이후 관리자가 아티팩트를 수정함으로써 새로운 아티팩트를 수집할 수 있는 상태가 된다. Windows 환경에서 DropBox 앱 사용 흔적 조사 방법을 제시한 연구[9]에서는 Windows의 파일 시스템인 NTFS에서 시간 값이 변화하는 행동을 확인하여 DropBox 앱 사용 흔적을 추정하였다. DropBox 내의 아티팩트에 저장되어 있는 사용자 계정 값과 사용자 행동에 따른 데이터들을 수집하여 NTFS에서와 DropBox 내에서의 흔적을 조사하였다. 클라우드 드라이브의 포렌식 데이터 획득 방안을 제시한 연구[10]는 클라우드 자체에서 지원하는 API를 통하여 데이터를 획득하는 도구를 개발하였다. 클라이언트에서 클라우드 데이터를 획득하는 것과는 달리, 클라우드 스토리지에 직접 API를 이용하여 접근하고 메타데이터를 직접 얻는 것으로 증거를 신뢰할 수 있다고 설명하였다. 먼저 크리덴셜을 획득하여 해당 도구에 크리덴셜을 부여한 뒤, 클라우드 스토리지 내의 파일 목록을 csv로 출력하였으며, 파일의 메타데이터를 JSON 형태로 출력하여 획득하였다. 위 연구들은 본 논문과 유사하게 클라우드 아티팩트를 분석하여 사용자의 행위, 업로드된 파일의 정보 등을 수집하는 방안을 제시하였다. 그러나 아티팩트뿐만 아니라 크리덴셜을 획득하여 활용할 수 있는 방안을 제안하는 부분에서 차이가 존재한다.

2.1 분석환경

본 논문에서는 모바일 클라우드 스토리지 앱 분석을 위해 분석환경을 다음과 같이 구축하였다. 앱의 아티팩트 및 데이터 분석은 HxD[11]를 통해 이루어졌다. 데이터베이스의 테이블 및 컬럼 분석은 DB Browser for SQLite[12]를 사용하였다. 데이터베이스 생성에 필요한 사용자 계정 및 암호는 JEB Decompiler[13]를 통해 앱의 바이트코드를 Java 소스코드로 디컴파일 하여 분석하였다. 앱의 아티팩트 데이터와 크리덴셜을 획득하기 위해 루팅된 모바일 기기를 사용하였다. 본 논문에서 사용한 도구와 앱의 패키지 명에 대한 정리는 다음 <Table 1>과 같다.

〈Table 1〉 Analysis Environment

Device and Software	Version	
Pixel 4a	Android 11	
HxD (hex deitor)	2.4.0.0	
DB Browser for SQLilte	3.12.0.0	
JEB Decompiler	3.7.1	
Application	Package	Version
TeraBox	com.dubox.drive	2.10.3
DeGoo	com.degoo.android	1.57.167.220202
Yandex.Disk	ru.yandex.disk	5.18.0

III. TeraBox

TeraBox는 바이두 재팬의 자회사 Flextech Inc.가 운영하는 클라우드 스토리지다[14]. 1TB의 용량을 무료로 제공하며 클라우드 스토리지에서 토렌트를 사용 할 수 있는 기능을 제공하고 있다. 클라우드 스토리지 내에 있는 파일을 링크를 생성하여 링크소유자에게 파일을 공유할 수 있는 기능이 있으며, 화면 잠금, 보안 폴더 같은 개인 프라이버시 보안 기능도 함께 제공하고 있다. 보안 폴더는 코퍼라는 이름으로 제공된다. 코퍼는 사용자가 패스워드를 등록하여 활성화한다. 패스워드의 범위는 0000-9999까지 4자리 숫자로 이루어져 있으며 패스워드 복구 및 변경 기능을 제공하지 않아 사용자가 패스워드를 변경을 시도하면 기존에 저장되어있던 파일이 모두 삭제된다. 본 장에서는 TeraBox의 아티팩트 정보들과 크리덴셜의 획득 및 재사용에 대해 설명한다.

3.1 크리덴셜 획득

TeraBox의 로그인은 바이두의 OAuth를 통해 이루어진다. 인증 과정을 통해 두 가지 토큰 bduss, uid를 발급받게 되며 이를 활용해 로그인이 진행된다. 두 토큰 값은 data/data/com.dubox.drive/databases/account.db의 info테이블에 저장된다. bduss는 Base64 인코딩된 28-byte의 문자열로 이루어져 있으며 account_bduss 컬럼에 저장되고, uid은 사용자 고유값으로 11자리의 숫자로 account_uid 컬럼에 저장된다. 동적분석 결과 실제로 두 값이 로그인 API에 활용되는 것이 확인됐다.

The image shows a file explorer interface with a list of database files on the left and a table view of account data on the right. The table has columns for id, account_uid, account_name, account_phone, account_email, and account_bduss. The first row shows a filter applied to account_uid.

id	account_uid	account_name	account_phone	account_email	account_bduss
1	20022723216	[REDACTED]	[REDACTED]	[REDACTED]	YqYOM6eteHu4S100206TqxxE5wTZFkMolgh...

Below the table, the text "Credential Value" is visible. The file list on the left includes files like _config.db, 20022723216_cloudimage.db, 20022723216_config.db, 20022723216_homecard.db, 20022723216_message.db, 20022723216_safebox.db, 20022723216_sharelink.db, 20022723216_sharesource.db, 20022723216backup.db, 20022723216filelist.db, 20022723216photo_album_backup.db, 20022723216resources.db, 20022723216video_album_backup.db, account.db (highlighted), ads.db, google_app_measurement_local.db, log.db, stats.db, video_preload_cache.db, and webcache.db.

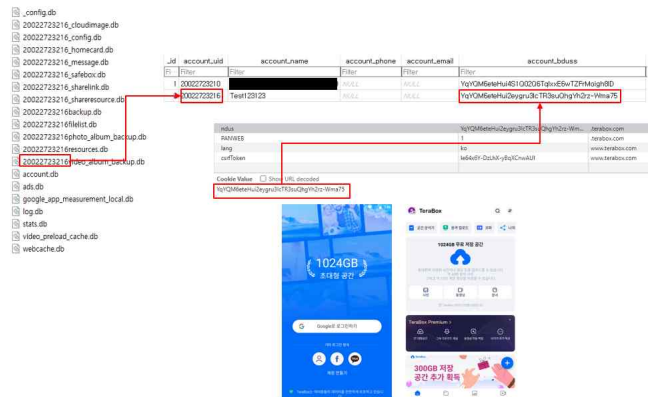
〈Figure 1〉 TeraBox's Credential Value

데이터베이스에 저장된 사용자 크리덴셜 값을 획득하여 다른 기기에 추가하는 경우, 해당 사용자로 로그인할 수 있다(Figure 2). 획득한 크리덴셜은 원본 기기 혹은 크리덴셜을 재사용한 기기 중 어느 한 곳에서 로그아웃하기 전까지 사용가능하며, 로그아웃을 하는순간 만료되며 로그아웃된다. 하지만 로그아웃 이후 사용되었던 크리덴셜이 남아있는 경우 다시 ID와 패스워드 없이 재 로그인이 가능하다.



〈Figure 2〉 TeraBox's Credential Reuse

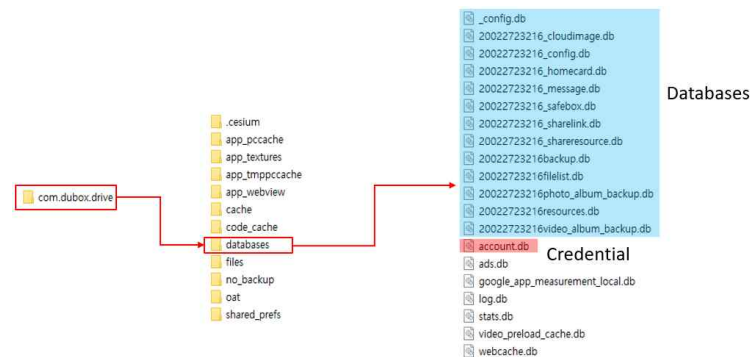
TeraBox 앱의 크리덴셜은 TeraBox 웹서비스에서 로그인할 경우 “ndus” 라는 쿠키값으로 저장된다. 해당 값과 uid값을 account.db 내에 데이터를 추가할 경우 로그인이 가능하다(Figure 3). uid의 경우 로그아웃을 할 경우에도 databases 폴더 내에 파일명이 uid인 db들이 남아있다. 이를 이용하여 획득한 uid와 ndus 쿠키값, 임의의 유저명을 추가하면 해당 크리덴셜 사용자의 클라우드 스토리지에 접근할 수 있다. 하지만 앱에서 얻은 크리덴셜 값을 이용하여 웹으로 접근은 불가능하다.



〈Figure 3〉 TeraBox's Credential Restore

3.2 아티팩트 분석

TeraBox 데이터의 구조는 다음 〈Figure 3〉와 같다. 패키지 명은 com.dubox.drive 이며, /data/data/com.dubox.drive 하위 경로에 클라우드 사용 기록, 저장 경로 등 다양한 아티팩트가 평문상태로 저장된다.



〈Figure 3〉 TeraBox's Data Structure

databases 파일들에는 사용자 데이터가 저장되어있다. 각 db는 중복되는 데이터가 있거나 데이터가 남지 않는 파일이 있으며, 본 논문에서는 증거로 사용될 수 있는 데이터가 남는 db를 분석하였다. {UID}filelist.db는 사용자가 모바일 앱을 이용하여 클라우드에 저장한 파일과 목록, 클라우드 사용 행위들이

저장되어있다. 그중 album_backup 테이블은 사용자가 백업을 등록해둔 데이터의 해시값과 원본 경로, 클라우드 스토리지 경로, 파일 크기, 업로드한 시간이 기록된다(Figure 4).

_id	local_url	remote_url	size	date
Filter	Filter	Filter	Filter	Filter
1	/storage/emulated/0/Pictures/Screenshots/Screenshot_20220110-134807.png	/Screenshot_20220110-134807.png	60201	1641790087000
2	/storage/emulated/0/Pictures/Screenshots/Screenshot_20211215-172754.png	/Screenshot_20211215-172754.png	524763	1639556874000
3	content://com.android.providers.media.documents/document/document%3A702	/chatting20220102_232653.txt	206	1641133613000

〈Figure 5〉 album_backup Table

cashfilelist 테이블에는 사용자가 TeraBox를 이용하여 업로드한 행위들이 기록된다(Figure 5). 사용자가 업로드한 파일의 id, 클라우드 스토리지에서의 경로, 저장된 파일의 이름 또는 폴더의 이름, 파일을 저장한 시간이 각각 서버와 클라이언트의 시간으로 저장된다. 파일의 경우에는 파일의 MD5 해시값이 저장된다.

server_path	file_name	isdir	state	file_category	file_property	parent_path	blocklist	file_md5
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
/From : Pixel 4a/DCIM	DCIM	1	0	6	0	/From : Pixel 4a/	NULL	
/From : Pixel 4a/DCIM/...	20211129_171736_446.jpg	0	0	3	0	/From : Pixel 4a/DCIM/	NULL	2821e0b8d9cc26816015e0e0cc1cc8de
/From : Pixel 4a/DCIM/Camera/...	PXL_20210727_044626766.jpg	0	0	3	0	/From : Pixel 4a/DCIM/Camera/	NULL	d8a4c7abe95b31ba48d9f7b546551
/From : Pixel 4a/DCIM/Camera	Camera	1	0	6	0	/From : Pixel 4a/DCIM/	NULL	
/From : Pixel 4a/DCIM/...	20211129_171736_549.jpg	0	0	3	0	/From : Pixel 4a/DCIM/	NULL	a10e0971bw74b6507ca0a0c89fcbcd0
/From : Pixel 4a/DCIM/검은색1.png	검은색1.png	0	0	3	0	/From : Pixel 4a/DCIM/	NULL	42439e22q154ea362bc0e457767a5e
/From : Pixel 4a/DCIM/Camera/...	PXL_20210723_080945818.jpg	0	0	3	0	/From : Pixel 4a/DCIM/Camera/	NULL	a1deab937hae9e1b59949067df637943

〈Figure 6〉 cashfilelist Table

v_record_files 테이블에는 사용자가 업로드한 파일의 정보들이 좀 더 자세하게 저장되어있다. 파일의 경로, 파일의 이름, 디렉토리여부, 파일의 확장자, 원본 경로, 파일의 MD5 해시값, 파일의 크기, 업로드된 시간이 저장되어있다. {UID}filelist.db에 저장되는 주요 데이터를 정리하면 다음 (Table 2)와 같다.

〈Table 2〉 Table And Column In {UID}Filelist.db

Table	Column Name	Data	Remark
album_backup	_id	Index Value	Index
	local_url	Local File Path	File Path
	remote_url	Server URL Path	URL Path
	size	Server Size	Byte
	date	Backup Date	Unix Time (Milliseconds)
cashfilelist	_id	File Index Value	
	fid	File ID Value	
	server_path	Server URL Path	
	file_name	File Name	
	isdir	Value of Directory	0 : File 1 : Directory
	file_category	File Extension Value	1 : Video File 3 : Picture File 4 : Text File 6 : Other
	parent_path	file_category	
	file_md5	File Length	File Size
	file_size	Delete Time	System Time
	server_ctime	Server Change Time	Unix Time (Milliseconds)
	server_mtime	Server Modify Time	
	client_ctime	Client Change Time	
	client_mtime	Client Modify Time	

v_record_files	id	File Add Time	Unix Time (Milliseconds)
	date		
	c_time_millis	Change Time	
	record_id	File Add Time	
	fsid	File Identity Value	
	_id	File Index Value	
	fid	File Identity Value	
	server_path	Server URL Path	
	file_name	File Name	
	isdir	Value of Directory	0 : File 1 : Directory
	file_category	File Extention Value	1 : Video File 3 : Picture File 4 : Text File 6 : Other
	parent_path	File Parent Path	
	file_md5	File MD5 Hash Value	MD5 Hash
	file_size	File Size	Byte
	server_ctime	Server Change Time	Unix Time (Milliseconds)
	server_mtime	Server Modify Time	
	client_ctime	Client Change Time	
	client_mtime	Client Modify Time	

{UID}sharelink.db는 TeraBox를 이용하여 파일을 공유할 때 생성되는 링크가 저장되어있다. 해당 파일의 경로와 파일의 종류, 설정한 기간과 링크가 저장되어있다. 해당 파일 공유 링크를 획득하면 URL에 접속하여 해당 링크로 공유된 데이터를 획득할 수 있다. 공유 링크의 기간을 설정하지 않으면 0으로 기록되며 영구히 링크가 존재한다(Figure 6).

id	typical_path	typical_category	expired_time_second	c_time	short_link
Filter	Filter	Filter	Filter	Filter	Filter
399062601	/Screenshot_20220110-134807.png	3	0	1642563757	https://terabox.com/s/1LHrPSuZmB2Xglsbntcm-yg
2108853806	/PXL_20220119_033258492.mp4	1	0	1642563748	https://terabox.com/s/1UqNdY9s5Kb1O9PpxJs5g
3676730867	/Screenshot_20211215-172754.png	3	1643173493	1642566693	https://terabox.com/s/1QosVdRiZu907Kj1KVBYQw
4210619403	/PXL_20220119_033258492.mp4	1	0	1642566681	https://terabox.com/s/1Al4nkC5JyQeZ7M8nJHwQ

〈Figure 7〉 File Share Link Within {UID}sharelink.db

TeraBox의 보안 폴더 기능인 코퍼는 safebox.db에 정보가 저장되어있다. 데이터베이스에는 해당 코퍼에 업로드한 파일의 이름, 파일 종류, 파일의 경로, 디렉토리여부, 파일의 종류가 저장되어있다(Figure 7).

_id	fid	server_path	file_name	isdir	state	file_category	file_property	parent_path	blocklist
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
14	362847311595536	/_pcs_safebox/...	Screenshot_20220110-134807.png	0	0	3	0	/_pcs_safebox/	NULL
15	52377306985713	/_pcs_safebox/...	Screenshot_20211215-172754.png	0	0	3	0	/_pcs_safebox/	NULL
16	968616181797943	/_pcs_safebox/PXL_20210826_090124979.mp4	PXL_20210826_090124979.mp4	0	0	1	0	/_pcs_safebox/	NULL

〈Figure 8〉 Column In safebox.db

사용자 계정 정보와 크리덴셜은 account.db파일 내의 info 테이블에 저장된다(Table 3). 데이터베이스 내에는 사용자의 uid와 사용자이름, 가입된 핸드폰 및 이메일주소, bduss 크리덴셜, 화면 잠금 패스워드가 저장되어있다. 화면 잠금 패스워드는 4자리의 패스워드로 이루어져 있으며 숫자만 입력할 수 있다. 패스워드는 MD5 (Message Digest algorithm 5) 해시로 한번 해싱되어 lock_password 컬럼에 저장된다. 잠금 기능은 is_lock_password_enable 컬럼의 값을 Disable에 해당하는 0 값으로 변경하여 기능을 해제할 수 있다.

〈Table 3〉 info Table Column In account.db


Table	Column Name	Data	Remark
info	_id	Index Value	
	account_uid	User UID	
	account_phone	User Phone Number	
	account_email	User Email Address	
	account_bduss	bduss Cookie	BDUSS Cookie Value
	is_current_login	Login Status Value	0 : Logout 1 : Login
	lock_password	Screen Lock Password	MD5 Hash Value
	is_lock_password_enable	Screen Lock Status	0 : Disable 1 : Enable
	nick_name	User Name	
	avater_url	User Avater URL	
	cur_country	User Contry	

IV. Yandex.Disk

Yandex.Disk는 러시아 최대 검색엔진 기업 Yandex에서 운영하는 클라우드 스토리지 서비스다[15]. Yandex.Disk는 무료 용량 10GB를 제공하고 있다. 파일을 삭제할 시 휴지통 기능을 제공하며, 클라우드 스토리지 앱에서 메일 기능도 함께 사용할 수 있다. 보안 기능으로는 PIN을 이용한 화면 잠금 기능이 제공된다.

4.1 크리덴셜 획득

Yandex.Disk의 크리덴셜은 /data/data/ru.yandex.disk/databases 폴더 내에 PassportInternal.db 파일 내에 사용자 정보와 함께 파일에 저장되어있다. 크리덴셜은 master_token_value 컬럼에 Base64로 인코딩된 값 39-byte의 문자열로 저장되어있다.



name	master_token_value	uid
Filter	Filter	Filter
[REDACTED]	AQAAAABcsGKrAAVs4CjcBRdB0gAlx-kbQ8w304	1:1555063467
[REDACTED]	AQAAAABcsGyEAAVs1762NihA0zNmGokStw1f4Q	1:1555065988

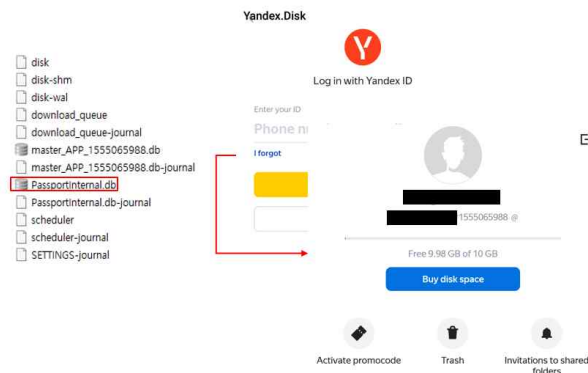
〈Figure 9〉 Yandex.Disk's Credential Value

사용자 계정 정보로는 사용자의 이름과 UID, 로그인 타입, 아바타 URL이 저장되어있으며 종합적인 정보는 데이터베이스 내에 JSON 형태로 저장되어있다〈Table 4〉. 로그인 타입은 지원되는 소셜미디어 계정인 구글, Facebook과 같은 계정으로 접근할때 social로 기록되며, Yandex의 계정은 login으로 기록된다.

〈Table 4〉 Column In PassportInternal.db

Table	Column Name	Data	Remark
info	name	User Name	
	master_token_value	Credential Data	Credential Token
	uid	User ID Value	
	user_info_body	User Info Data	
	user_info_meta		
	legacy_account	User Login Type	social : Social Media Login login : Yandex Login
	legacy_extra_data_body	User Info Data	JSON Extra Data

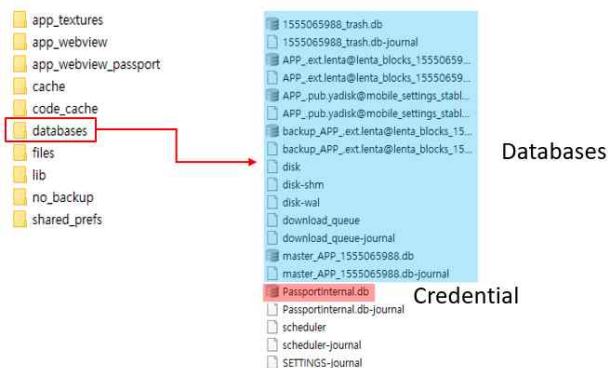
Yandex.Disk의 크리덴셜은 전용 API를 통해 암호화된 크리덴셜값을 서버에서 전송받는다. 안드로이드의 AccountManager 기능을 이용하여 해당 계정의 AuthToken값을 서버로 전송하면 발급되는 형식을 사용한다. 따라서 해당값을 획득 할 경우 기기에 상관없이 타 기기에서도 재사용이 가능하다. 따라서 해당 사용자의 크리덴셜 값을 획득하여 다른 기기에 적용한다면, 해당 사용자로 로그인할 수 있다(Figure 9).



〈Figure 9〉 Yandex.Disk Credential Reuse

4.2 아티팩트 분석

Yandex.Disk의 패키지명은 ru.yandex.disk이며 /data/data/ru.yandex.disk 경로에 데이터가 존재한다. 클라우드 파일 업로드 기록 및 휴지통 기록은 databases 폴더 내에 저장되어있다(Figure 10).



〈Figure 10〉 Yandex.Disk's Data Structure

자동 업로드 파일에 대한 정보는 확장자가 없는 SQL DB 파일인 disk 내에 존재하며 SQL DB 파일이다. AUTOUPLOADED_MOMENT_ITEMS1과 2 컬럼에는 업로드된 파일명, 시간, 경로 등이 저장된다. DISK_VIEW 컬럼에는 클라우드에 저장된 파일의 이름과 크기, 경로, 수정 날짜, 확장자가 저장되어있다. MediaHashes 컬럼에는 파일의 MD5 해시값과 SHA-256 (Secure Hash Algorithm 256byte) 해시값이 저장되어있다. DISK_QUEUE_EXXT 컬럼에는 각 파일을 어떤 네트워크에서 업로드했는지와 업로드된 경로, 시간이 포함되어있다. disk 파일 내의 주요 테이블 및 컬럼의 데이터는 다음 〈Table 5〉와 같다.

〈Table 5〉 Table And Column In disk

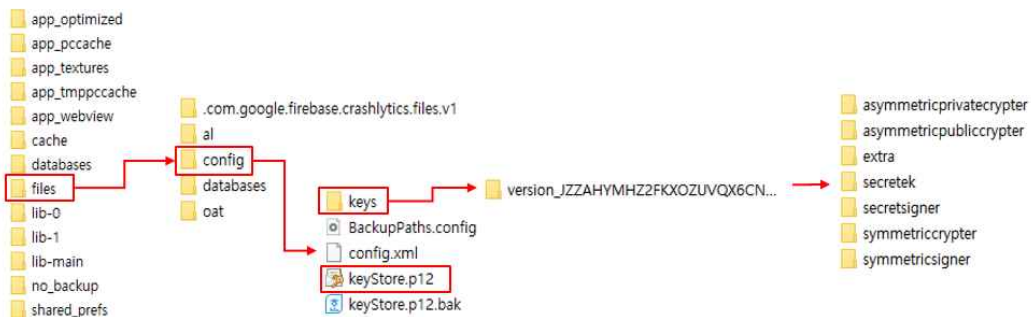
Table	Column Name	Data	Remark
AUTOUPLOADED_MOMENT_ITEMS1 or 2	uploaded_time	File Uploaded Time	Unix Time (Milliseconds)
	user	User ID	Nickname # UID
	from_autoupload	Auto Upload Value	1 : Auto 0 : Static
	PARENT	Parent File Path	
	NAME	File Name	
	RESOURCE_ID	User and File ID	UID:File ID
	DISPLAY_NAME	Displayed Name	
	DISPLAYED_NAME_TOLOWER	Displayed Lower Name	
	SIZE	File Size	Byte
	IS_DIR	Value of Directory	0 : File 1 : Directory
	LAST_MODIFIED	Last Modified Time	Unix Time (Milliseconds)
	MIME_TYPE	File Extention	
	SHARED	File Shared Value	0 : None 1: Shared
	READONLY	Read Only Attribute	0 : None 1 : ReadOnly
	MEDIA_TYPE	Media File Type	image video
	EXTENSION	File Extention String	
DISK_VIEW	_id	Index Value	
	PARENT	Parent File Path	
	NAME	File Name	
	RESOURCE_ID	User and File ID	UID:File ID
	DISPLAY_NAME	Displayed Name	
	DISPLAYED_NAME_TOLOWER	Displayed Lower Name	
	SIZE	File Size	Byte
	IS_DIR	Value of Directory	0 : File 1 : Directory
	LAST_MODIFIED	Last Modified Time	Unix Time (Milliseconds)
	MIME_TYPE	File Extention	
	SHARED	File Shared Value	0 : None 1: Shared
	READONLY	Read Only Attribute	0 : None 1 : ReadOnly
	MEDIA_TYPE	Media File Type	image video
	HAS_THUMBNAIL	Thumbnail Attribute Value	0 : None 1 : Has Thumbnail
	LAST_ACCESS	Last Access Time	Unix Time (Milliseconds)
	ROW_TYPE	Row file or not	None : NULL Row File : 1
	EXTENSION	File Extention String	
DISK_QUEUE_EXT	upload_id	Upload Index Value	
	uploaded_path	File Uploaded Path	
	added_to_queue_time	File Upload Added Time	Unix Time(Milliseconds)
	upload_started_time	File Upload Started Time	Unix Time(Milliseconds)
	enqueued_network	Network Type	wifi : WIFI-Network NULL : Other Network
MediaHashes	path	File Path	
	mTime	Modify Time	Unix Time(Milliseconds)
	size	File Size	Byte
	md5	File MD5 Hash	MD5
	sha256	File SHA256 Hash	SHA256

V. DeGoo

DeGoo는 2012년도에 설립된 스웨덴회사인 DeGoo Backup AB에서 제작한 앱이다[16]. DeGoo는 100GB의 용량을 무료로 제공한다. 유료 결제 시 년 180,000원의 결제로 암호화 기능과 용량을 추가하여 사용할 수 있다. 기본적으로 제공되는 기능으로는 클라우드 스토리지 서비스, PIN 화면 잠금, 앨범 정리 등을 제공한다. 모바일 웹 기반 앱이며 PC 앱을 이용하여 업로드할 경우 자동으로 7일 뒤에 삭제된다. 본 절에서는 DeGoo의 아티팩트 분석 및 크리덴셜 파일에 대해 설명한다.

5.1 크리덴셜 획득

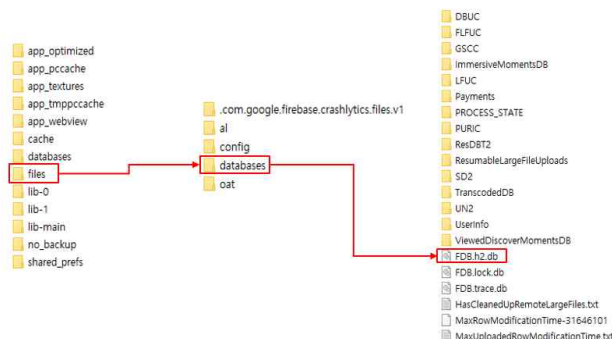
DeGoo의 크리덴셜 파일은 data/data/com.degoo.android/files/config 폴더 내에 저장되어있다. 사용자의 대칭키와 인증서 및 암호화키가 저장되어있다(Figure 11). keyStore.p12 파일과 keys 폴더에 있는 암호화키 파일들이 DeGoo의 크리덴셜 파일이다. KeyStore.p12는 PKCS (Public-Key Cryptography Standard)#12로 저장되어 있다. 해당 크리덴셜은 서버와의 통신에서 클라이언트 인증을 위해 사용된다. DeGoo는 사용자와 서버와의 세션을 SSL로 생성한다. 이때 사용된 인증서를 PKCS#12 형태로 저장한다. 해당 인증서는 인가된 사용자를 인증할 때 사용하며, Android KeyStore에 저장되어있는 키로 암호화해 저장한다. 따라서 타 기기에는 재사용하기 어렵다. keys 폴더는 서버와 연결할 때 사용하는 대칭키 또는 비대칭키가 저장되어있다. keys에 있는 키들과 인증서인 keyStore.p12파일이 함께 존재하여야 해당 계정에 접근할 수 있다.



〈Figure 11〉 DeGoo's Credential Files

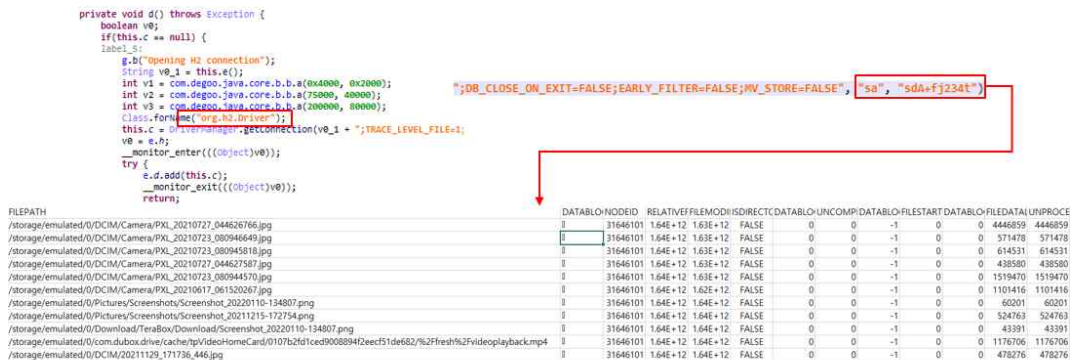
5.2 아티팩트 분석

DeGoo의 패키지 명은 com.degoo.android 이며 /data/data/com.degoo.android 하위 경로에 앱 데이터가 저장된다. 모든 데이터는 암호화되어 있지 않은 상태로 저장된다. DeGoo는 files 폴더 내에 databases 폴더에 사용자 정보와 클라우드 스토리지 사용 기록 등이 저장되어있다. DeGoo의 데이터베이스는 h2 데이터베이스[17] 파일로 존재한다(Figure 12).



〈Figure 12〉 DeGoo's Data Structure

해당 h2 데이터베이스는 데이터 확인 시 계정명과 패스워드를 사용한다. 해당 값은 Dalvik Executable 파일에 고정값으로 저장되어있다. 정적 분석 결과 “sa”와 “sdA+fj234t” 두 문자열이 하드코딩 되어 있었으며 각각 계정명과 패스워드에 사용된다(Figure 13). 해당 데이터베이스를 추출할 경우, 업로드한 기록이 저장되어있으며, 업로드한 파일의 원본 경로, 디렉토리여부, 파일 삭제 여부, 파일의 체크섬 값이 저장되어있다.



〈Figure 13〉 Extracting h2 Database Data

사용자 정보와 기기 정보는 패키지명 하위 /files/databases 폴더 내에 Userinfo 폴더와 UN2 폴더에 저장되어있다. 해당 파일은 확장자가 없으며 임의의 숫자값으로 파일명이 생성된다. 내부 데이터는 hex 값을 직접 확인하여 데이터를 획득할 수 있다. 암호화는 되어있지 않으며 저장되어있는 정보는 사용자 계정, 활성화 여부, 기기 정보가 저장되어있다(Figure 14). 해당 /files/databases 폴더 하위에 Userinfo와 UN2 폴더를 제외한 나머지 폴더들에는 숫자 값으로 생성된 파일들이 저장되어있으며 의미 있는 데이터는 저장되어있지 않다.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 0A 12 0A 10 75 73 65 72 5F 69 6E 66 6F 5F 64 62 ...user info db
00000010 5F 6B 65 79 12 62 08 80 DB B6 AA E7 2F 12 59 7B key.b.€Üq*c/.Y(
00000020 22 61 22 3A 22 "a":
00000030 "b":
00000040 22 3A 38 36 39 32 33 34 2C 22 63 22 3A 31 30 "8692334,"c":10
00000050 37 33 37 34 31 38 32 34 30 30 2C 22 64 22 3A 22 7374182400,"d":
00000060 46 72 65 65 22 2C 22 65 22 3A 22 55 6E 61 76 61 Free,"e":"Unava
00000070 69 6C 61 62 6C 65 22 7D ilable")

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 0A 05 08 95 C3 8B 0F 12 A6 01 0A 05 08 95 C3 8B ...*Ä...*Ä
00000010 0F 12 05 08 DE F8 F3 0C 1A 0F 47 6F 6F 67 6C 65 ...*ø...Google
00000020 20 50 69 78 65 6C 20 34 61 29 00 00 00 00 00 00 Pixel 4a).....
00000030 00 00 31 00 00 00 00 00 00 00 00 00 00 00 00 00 .l.....9....
00000040 00 00 00 00 40 E0 DA B6 AA E7 2F 4A 04 08 00 10 ...@âÜq*c/J....
00000050 00 52 17 .R.
00000060 5A 30 EC B5 9C EC 201pxel
```

〈Figure 14〉 DeGoo's User Data

VI. 결 론

〈Table 6〉 Study Summary Table

App	Credential Location	Credential Features	Credential Reuse Conditions
TeraBox	./databases/account.db, Web ndus Cookie	OAuth Token	Rooting Required
DeGoo	./files/config/keyStore.p12	PKCS#12 Certification File	Rooting Required/ Keystore Access Required
Yandex.Disk	./databases/PassportInternal.db	Encoded Base64 Token data	Rooting Required

본 논문에서는 모바일 클라우드 서비스 앱인 TeraBox, DeGoo, Yandex.Disk에 대한 아티팩트와 크리덴셜을 분석하였다. 앱의 크리덴셜은 각각 데이터베이스와 앱 데이터 내에 저장되어있었으며 앱마다 OAuth Token, PKCS#12 형태의 인증서파일, 인코딩된 Base64 토큰데이터로 저장되어있었다(Table 6). 앱에 따

라 크리덴셜 재사용 여부에 차이가 존재하였다. 크리덴셜 재사용 조건은 루팅된 기기가 필요하거나 키스토어 접근이 필요하다. 앱 아티팩트 데이터는 사용자가 클라우드를 사용한 행위 기반으로 데이터가 생성되었으며, 업로드 기록과 시간, 업로드한 네트워크, 파일의 해시값과 같은 다양한 데이터들이 저장되었다. 이러한 데이터들은 디지털 포렌식 수사에서 용의자의 행위 또는 범죄 특징에 도움을 줄 수 있으며, 잠금기능을 통해 앱의 접근을 하지 못하게 하여 수사를 방해할 경우 크리덴셜을 활용하는 방안을 제시하였다. 본 논문의 아티팩트 분석 및 크리덴셜 활용의 결과를 통해 분석된 모바일 클라우드 스토리지 앱의 데이터 수집 시간을 단축하고 크리덴셜 활용으로 디지털 포렌식 수사에 도움이 될 것으로 기대한다.

참 고 문 헌 (References)

- [1] Digital Daily, "[Weekly Cloud Trend] Cloud company flew with COVID-19", <https://www.ddaily.co.kr/news/article/?no=195139>
- [2] The Asia Business Daily, "71 percent of technology leaks, insiders do...Evolve techniques such as cloud abuse.", <https://www.asiae.co.kr/article/2021111610002962132>
- [3] Maeil Business, "[Exclusive] Another path to darkness..."Cloud sub-listening water." Dangerous level.", <https://www.mk.co.kr/news/society/view/2020/03/310998/>
- [4] SBS News, "You thought I wouldn't know?, A 20-year-old who hid illegal video clips in the cloud.", https://news.sbs.co.kr/news/endPage.do?news_id=N1005432671
- [5] Incheon Ilbo, "'Illegal shooting' Incheon professional team staff said, 'I forgot the password'. Investigation difficulties.", <http://www.incheonilbo.com/news/articleView.html?idxno=1109097>
- [6] Hyemin Yun, Jaeuk Kim, Eunbi Hwang, Haeni Kim, Taekyoung Kwon."Analysis of Box Cloud Artifacts in Windows 10 Environments" REVIEW OF KIISC29,6(2019):29-37.
- [7] Dongho Kim, Sangjin Lee.(2018).A Study on the Usage of Investigation of Google Cloud Data (Smartphone user-oriented).Journal of Digital Forensics ,12(3),107-118.
- [8] Mingyu Kim, Doowon Jeong, Sangjin Lee. "The Automatic Collection and Analysis System of Cloud Artifact" Journal of the Korea Institute of Information Security & Cryptology 25, 6 (2015) : 1377-1383.
- [9] Ki-Uk Nam, Dong-Hyun Kim, Ji-sung Choi, Sang-jin Lee."Investigating traces of a Dropbox application for Windows"Journal of Digital Forensics 14,1(2020):45-58.
- [10] Vassil Roussev, Andres Barreto, and Irfan Ahmed, Forensic Acquisition of Cloud Drives, InAdvances in Digital Forensics XII, Gilbert Peterson and Sujeet Shenoi (eds.), Springer, 2016
- [11] HxD - Freeware Hex Editor and Disk Editor, <https://mh-nexus.de/en/hxd/>
- [12] DB Browser for SQLite, <https://sqlitebrowser.org/>
- [13] JEB Decompiler by PNF Software, <https://www.pnfsoftware.com/>
- [14] Terabox: Cloud Storage Space, <https://www.terabox.com/>
- [15] Degoo Cloud - Life's best memories, <https://degoo.com/about>
- [16] Wikipedia, the free encyclopedia, <https://en.wikipedia.org/wiki/Yandex>
- [17] H2 Database Engine, <https://www.h2database.com/html/main.html>

저 자 소 개



최 용 철 (Yongcheol Choi)

준회원

2021년 2월 : 서원대학교 정보보안학과 졸업

2021년 3월 ~ 현재 : 국민대학교 금융정보보안학과 석사과정

관심분야 : 디지털 포렌식, 정보보호 등



김 기 윤 (Kiyoon Kim)

준회원

2019년 2월 : 국민대학교 수학과 졸업

2019년 3월 ~ 현재 : 국민대학교 금융정보보안학과 석박통합과정

관심분야 : 디지털 포렌식, 정보보호 등



김 중 성 (Jongsung Kim)

평생회원

2006년 11월 : K.U.Leuven, ESAT/SCD-COSIC 정보보호 공학박사

2007년 2월 : 고려대학교 정보보호대학원 공학박사

2009년 9월 ~ 2013년 2월 : 경남대학교 e-비즈니스학과 교수

2013년 9월 ~ 2017년 2월 : 국민대학교 수학과 교수

2017년 3월 ~ 현재 : 국민대학교 정보보안암호수학과/금융정보보안학과 교수

관심분야 정보보호, 암호 알고리즘, 디지털 포렌식