Analyse Forensique Android :

Récupération et Analyse des Données Supprimés

Rapport rédigé par :

Jean Arnaud ATEUMENA, Lamine DEMBELE

Mastère Spécialisé ECRSI

Télécom Paris

Digital_Forensic

Enseignant : M. François GONNET

03.06.2025

Sommaire

Table des images

1. Résumé Exécutif

Ce rapport présente une investigation forensique sur une image de test Android 11 (Google Pixel 3, archive logique .tar de la partition /data par Joshua Hickman), axée sur la "Récupération et analyse des données supprimées". L'étude a exploré les mécanismes de suppression sur Android, analysé les artefacts applicatifs et système, et démontré les techniques de récupération de données effacées, tout en reconstituant une chronologie des événements.

La méthodologie a inclus la vérification de l'intégrité de la source, son traitement via Autopsy, une analyse ciblée des bases de données de Google Photos (gphotos-1.db, local_media) avec un explorateur SQLite, et l'examen des logs système (logcat) de Magnet ACQUIRE. Une simulation de file carving sur une image disque dédiée a complété l'analyse.

Les découvertes majeures révèlent, dans Google Photos, des trash_timestamp (mise à la corbeille) correspondant à trois suppressions d'images documentées, ainsi que des purge_timestamp indiquant une politique de rétention de 30 jours, confirmant la "récupérabilité" des fichiers au moment de l'acquisition. Des différences de timestamps entre artefacts locaux et cloud (remote_media) ont souligné la complexité de la synchronisation. La base local_trash.db était vide et le logcat n'a pas fourni d'indices directs sur les suppressions. La simulation de file carving a permis de récupérer avec succès des images et documents.

En conclusion, cette étude confirme la persistance significative des métadonnées de suppression dans les applications Android. Elle souligne la complémentarité entre l'analyse d'artefacts applicatifs (pour le contexte et la chronologie) et les techniques de bas niveau comme le file carving (pour les données brutes, surtout sur images physiques). Les défis liés au type d'acquisition et au chiffrement, ainsi que l'importance d'une documentation contextuelle, ont été réaffirmés. Ce travail illustre la nécessité d'une approche méthodique et multi-facettes pour la récupération de données supprimées sur Android.

2. Introduction

2.1. Contexte de l'Exposé et Objectifs de l'Investigation

2.1.1. Contexte de l'Exposé

Dans le contexte numérique actuel, la criminalistique mobile, également appelée **mobile digital forensics**, occupe une place centrale dans les enquêtes modernes, qu'elles soient d'ordre pénal, cybercriminel ou organisationnel. Les smartphones, omniprésents dans notre quotidien, sont devenus de véritables coffres-forts numériques, renfermant des informations sensibles souvent déterminantes dans la résolution d'affaires judiciaires ou de cybersécurité.

La **réponse aux incidents numériques sur appareils mobiles (Mobile DFIR)** repose sur un ensemble de techniques visant à **collecter, analyser et préserver les données numériques** issues de téléphones, tablettes ou objets connectés. Ces données incluent entre autres les historiques d'appels, les messages, les e-mails, les activités sur les réseaux sociaux, les données de géolocalisation ou encore les traces d'utilisation des applications.

Cette discipline ne se limite pas aux forces de l'ordre : les entreprises et organisations y ont également recours pour enquêter sur des fuites d'informations, des fraudes internes ou encore des compromissions de systèmes. Dans tous les cas, l'objectif reste le même : **identifier les preuves exploitables tout en garantissant leur intégrité et leur recevabilité juridique.**

Parmi les environnements les plus fréquemment analysés figure **Android**, le système d'exploitation mobile le plus répandu dans le monde. L'**Android Forensics** consiste à extraire et interpréter les artefacts numériques générés par l'activité de l'utilisateur sur un appareil Android. Cela peut inclure des éléments comme les SMS, journaux d'appels, fichiers supprimés, historiques de navigation, bases de données applicatives ou encore journaux système.

L'analyse forensique d'un appareil Android requiert une **connaissance fine de son architecture**, ainsi que la maîtrise d'outils spécialisés capables de récupérer des données même sur des appareils verrouillés ou endommagés. L'expert en Android Forensics ne se contente pas d'extraire des données brutes : il doit également **analyser, recouper et contextualiser les informations** pour en tirer des conclusions fiables.

2.1.2. Objectifs de l'Investigation

Cet exposé, réalisé dans le cadre du module digital forensic, aborde le sujet ***"Récupération et analyse des données supprimées (partition de stockage, logcat)" sur Android***. Le but principal de cette investigation est d'explorer les mécanismes de suppression de données sur Android, d'analyser les artefacts applicatifs et systèmes susceptibles de conserver des traces de ces suppressions, et de démontrer comment ces traces, notamment les timestamps, permettent de reconstituer une chronologie des événements. Nous nous concentrerons en particulier sur l'analyse des artefacts de l'application Google Photos pour identifier et dater des suppressions d'images.

2.2. Présentation de la Source de Données

Pour mener à bien cette étude, nous avons utilisé une image de test Android 11 publiquement disponible, créée par Joshua Hickman (The Binary Hick). Cette image, nommée "Android 11 - Pixel 3 - Data.tar", représente une archive logique de la partition/data d'un appareil Google Pixel 3. La documentation détaillée fournie par J. Hickman, le PDF « Android 11 Image Creation Documentation » ainsi que les journaux d'acquisition de l'outil Magnet ACQUIRE v2.30.0.22097 (fichiers activity_log.txt et image_info.txt) ont été essentiels pour contextualiser l'acquisition et guider notre analyse.

Il est essentiel de noter que l'appareil a été rooté à l'aide de Magisk avant l'acquisition, permettant ainsi un accès complet à la partition **/data**. L'acquisition a été réalisée le 05 octobre 2020, et l'outil Magnet ACQUIRE a utilisé une méthode de streaming pour archiver le contenu de **/data** dans le fichier.tar, tout en effectuant un dump des logs système (logcat) en fin de processus. Les caractéristiques détaillées de l'appareil et de l'acquisition seront présentées dans la section Méthodologie comme première étape de l'analyse.

La nature logique de cette archive.tar (copie des fichiers et répertoires alloués) implique des limitations directes pour la récupération de données déjà supprimées avant la création de l'archive, car elle ne contient pas l'espace non alloué du stockage. Cette contrainte sera un point central de notre discussion. L'intégrité de l'archive a été vérifiée en comparant ses hachages avec ceux fournis dans la documentation d'acquisition.

Hash Values:

Android 11 – Pixel 3 – Data.tar

MD5	aca33f3de4e228c29bc39576865fe7d9
SHA1	a7a4b184946af72c97395462816931c453f073c4
SHA256	d4cb326a41cca5cc23a2c2cb1908498c7281b1b2fcc0f8781fb6b3945e6c815c

```
ubuntu@Lamine:~/forensic$ ls
'Android 11 - Pixel 3 - Data.tar'
ubuntu@Lamine:~/forensic$ md5sum Android\ 11\ -\ Pixel\ 3\ -\ Data.tar
aca33f3de4e228c29bc39576865fe7d9  Android 11 - Pixel 3 - Data.tar
ubuntu@Lamine:~/forensic$ sha256sum 'Android 11 - Pixel 3 - Data.tar'
d4cb326a41cca5cc23a2c2cb1908498c7281b1b2fcc0f8781fb6b3945e6c815c  Android 11 - Pixel 3 - Data.tar
ubuntu@Lamine:~/forensic$
```

Figure 1 : Vérification de l'intégrité de l'archive Data.tar par comparaison des hachages

2.3. Scénario d'Investigation

La documentation de création de l'image par Joshua Hickman détaille des actions utilisateur spécifiques, notamment la suppression d'images via l'application Google Photos le 2020-10-03, avec la mention "placed in trash". Notre investigation a ciblé ces actions pour identifier et analyser les artefacts correspondants dans les bases de données de Google Photos. Comme nous le détaillerons, l'analyse de la base de données gphotos-1.db a permis d'identifier des entrées pour des images avec des trash_timestamp (horodatages de mise à la corbeille) correspondant précisément aux heures des suppressions documentées. Cette découverte constitue une preuve directe de l'action de suppression et de la conservation de ces métadonnées par l'application.

En parallèle, nous aborderons les techniques de file carving via un scénario simulé pour illustrer la récupération de données depuis l'espace non alloué, une approche complémentaire nécessaire lorsque les métadonnées sont absentes ou que les fichiers ont été définitivement supprimés de la corbeille. La reconstitution de la chronologie des événements, basée sur ces timestamps et des logs système, sera un fil conducteur de notre analyse.

3. Cadre Théorique : Suppression et Récupération sur Android

3.1. Systèmes de Fichiers Android et Mécanismes de Suppression

3.1.1. Hiérarchie des répertoires

Partition de démarrage (/boot) : Au démarrage de votre téléphone, le chargeur de démarrage charge d'abord le noyau et le chargeur de récupération depuis la partition de démarrage. Le noyau est responsable des fonctions principales du système d'exploitation, et le chargeur de récupération vous permet de réinitialiser votre téléphone aux paramètres d'usine ou d'accéder à d'autres options avancées.

Partition système (/system) : La partition système contient les fichiers nécessaires au bon fonctionnement du système d'exploitation Android. Ces fichiers incluent les applications, les frameworks, les polices et les paramètres.

Partition de données (/data) : elle contient toutes les applications et données installées par l'utilisateur. Cela inclut les applications téléchargées, les photos, la musique, les vidéos et autres fichiers.

Partition de stockage (/storage) : cette partition contient des périphériques de stockage externes tels que des cartes SD ou des clés USB. Ces périphériques peuvent fournir de l'espace de stockage supplémentaire lorsque la mémoire interne de votre téléphone est saturée.

Partition de récupération (/recovery) : La partition de récupération est une partition spéciale qui permet de démarrer votre téléphone en mode de récupération ou chargeur de démarrage. Ce mode vous permet de réinitialiser votre téléphone aux paramètres d'usine ou d'accéder à d'autres options avancées.

Android utilise divers systèmes de fichiers pour ses partitions, les plus courants étant EXT4 et, plus récemment, F2FS (Flash-Friendly File System) pour la partition **/data**.

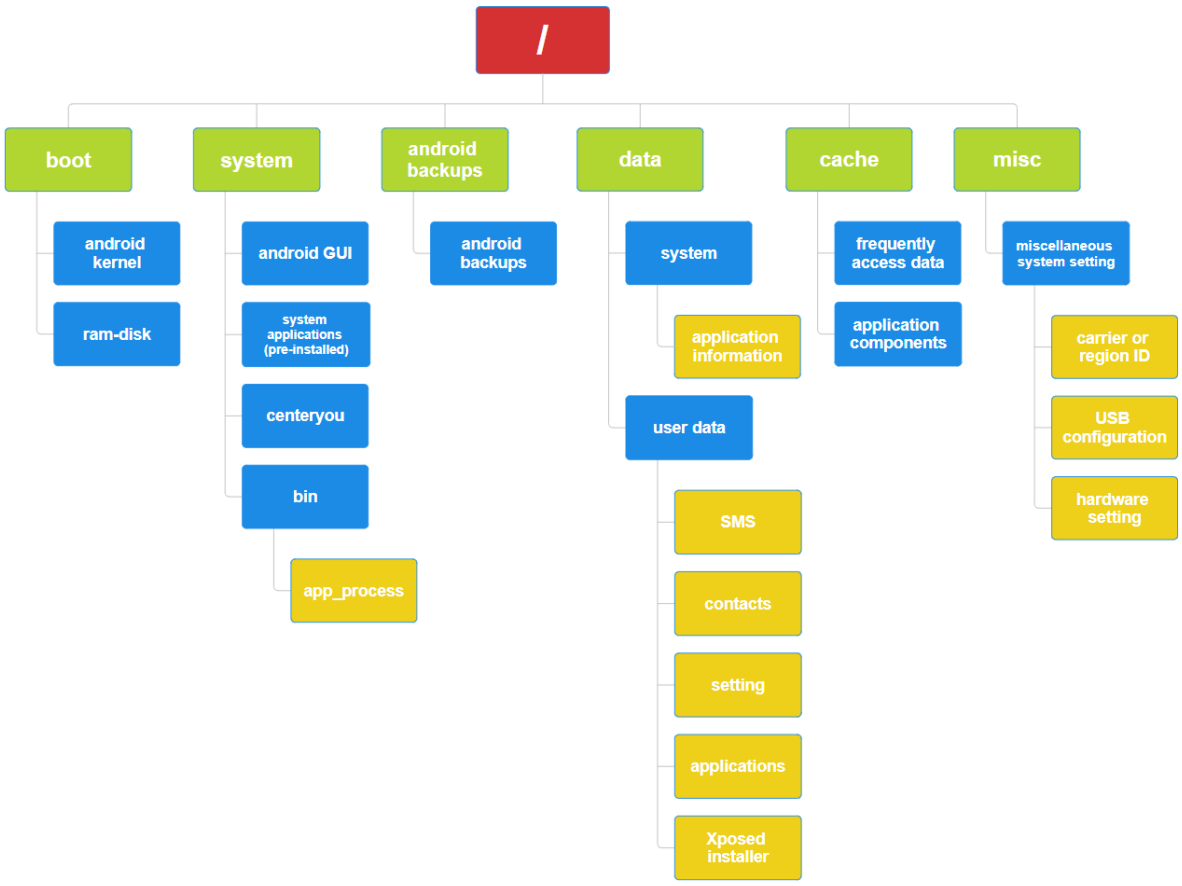


Figure 2 : Hiérarchie des répertoires d'un système Android

3.1.2. Mécanismes de Suppression

Lors de la suppression d'un fichier, le système d'exploitation ne procède généralement pas à un effacement immédiat des données sur le support de stockage. Typiquement, l'entrée du fichier dans la table du système de fichiers est marquée comme "supprimée" ou "libre", et les blocs de données qu'il occupait sont désignés comme réutilisables. Tant que ces blocs ne sont pas réécrits par de nouvelles données, le contenu original du fichier peut potentiellement subsister, rendant une récupération possible.

3.2. Impact du Chiffrement Basé sur les Fichiers (FBE)

Depuis Android 7 (Nougat), le chiffrement basé sur les fichiers (File-Based Encryption - FBE) est devenu la norme. Contrairement au chiffrement complet du disque (Full Disk Encryption - FDE), FBE chiffre les fichiers individuellement avec des clés différentes, certaines étant liées aux informations d'identification de l'utilisateur *(ex: code PIN, schéma)*. Le dataset utilisé

(Pixel 3 sous Android 11) est indiqué comme utilisant FBE *(cf. image_info.txt*). Bien que l'acquisition ait été réalisée sur un appareil déverrouillé (les fichiers dans le .tar sont donc déchiffrés), FBE complexifie significativement la récupération de données supprimée*s*, car les clés de chiffrement associées aux fichiers supprimés peuvent ne plus être accessibles, rendant les données récupérées potentiellement inutilisables même si les blocs sont retrouvés.

3.3. Techniques d'Acquisition : Logique vs. Physique

L'acquisition de données est une étape fondamentale. On distingue principalement :

- **L'acquisition physique (ou bit-à-bit) :** La méthode la plus fiable pour créer une image numérique forensique d'un appareil Android est de prendre une image physique. L'image physique est une copie bit à bit des zones mémoire de l'appareil et contient toutes ses données. Ce type d'image constitue la source de données la plus fiable et la plus complète pour les investigations forensiques. Cependant, il existe deux méthodes différentes pour créer une telle image, chacune présentant ses propres défis et risques :
    - La première méthode consiste à accéder au système en laboratoire à l'aide d'un matériel spécialisé pour copier les puces mémoire **(bloqueur d'écriture)**. Cette méthode requiert une expertise et un équipement de pointe et comporte le risque de causer des dommages irréparables au dispositif examiné.
    - Une autre approche d'acquisition physique, souvent plus accessible que la lecture directe des puces, consiste à accéder à l'appareil via **ADB**, puis potentiellement à installer des outils temporaires (agents, binaires) permettant d'opérer en tant qu'utilisateur root pour lire les partitions. Bien que cette méthode puisse impliquer l'écriture de ces outils temporaires sur l'appareil, ce qui constitue une modification à documenter soigneusement pour la chaîne de preuves, elle est généralement plus simple et rapide que la manipulation physique des puces. Il est à noter que des techniques d'acquisition physique encore plus pures, telles que **JTAG ou chip-off**, visent à lire directement la mémoire sans aucune écriture sur le support original, mais requièrent un équipement et une expertise plus spécialisés.

- **L'acquisition logique :** La deuxième option est l'imagerie logique, si une image physique ne peut être obtenue. L'imagerie logique consiste à connecter l'appareil à un ordinateur et à copier les fichiers du système de fichiers (extraction ciblée de fichiers /dossiers comme ce fut le cas pour notre archive.tar de /data). Bien que cette méthode soit relativement simple, elle offre un accès limité aux systèmes de fichiers (impossible d'accéder aux zones système) et permet uniquement de travailler avec les données de l'espace utilisateur. Une façon d'accéder aux fichiers est d'utiliser le protocole MTP.

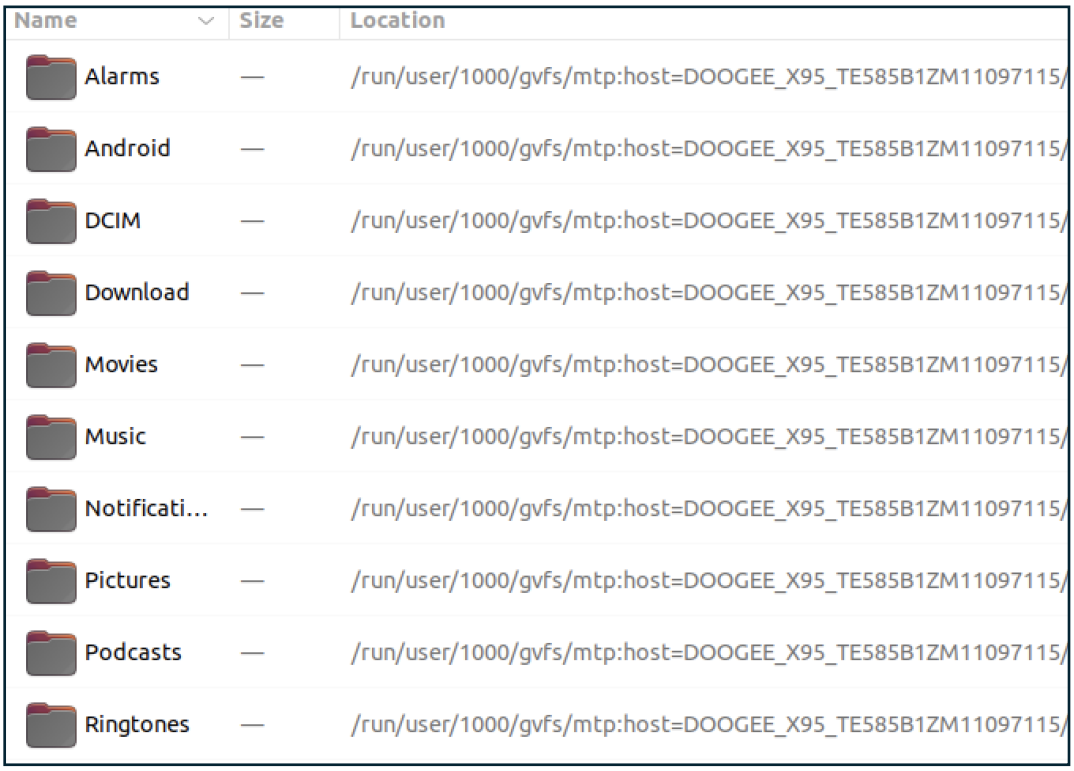


Figure 3: Imagerie logique

3.4. Principe du File Carving

Le file carving *(ou récupération par signature)* est une technique permettant de récupérer des fichiers de l'espace non alloué d'une image disque en se basant sur la reconnaissance de leurs structures internes, notamment leurs en-têtes (headers) et pieds de page (footers) spécifiques à chaque type de fichier (ex: FF D8 FF E0 pour JPEG, %PDF- pour PDF). Cette méthode est indépendante des métadonnées du système de fichiers et peut donc retrouver des fichiers même si leur référence a été complètement supprimée de la table d'allocation. Des outils comme PhotoRec ou les modules intégrés à des plateformes comme Autopsy

implémentent ces techniques. Les limites incluent la gestion des fichiers fragmentés et le risque de faux positifs.

3.5. Artefacts Applicatifs et Logs Système (logcat) comme sources d'indices

Au-delà de la récupération des fichiers bruts, l'analyse forensique sur Android s'appuie fortement sur les artefacts laissés par les applications et le système.

- **Artefacts Applicatifs :** Les applications stockent leurs données dans des répertoires spécifiques (généralement sous /data/data/<nom.du.package>/), souvent dans des bases de données SQLite, des fichiers XML (préférences partagées), des caches, etc. Ces artefacts peuvent contenir des métadonnées sur les fichiers gérés par l'application, y compris des références à des fichiers supprimés, des timestamps d'activité, ou des configurations.

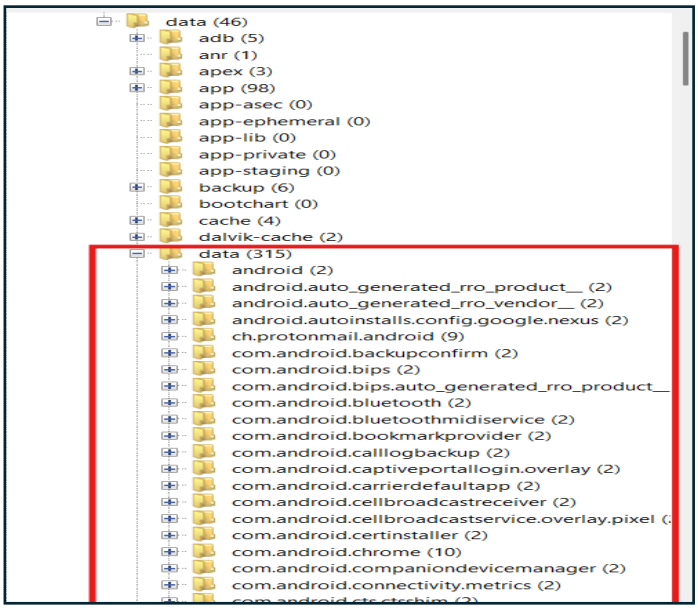


Figure 4: Exemple d'artefacts applicatifs

- **Logs Système (logcat) :** Android dispose d'un système de journalisation centralisé (logcat) qui enregistre les messages du système et des applications. Bien que logcat soit un buffer circulaire et que les logs persistants ne soient pas toujours disponibles ou complets, son analyse peut révéler des erreurs, des activités suspectes, ou des actions utilisateur (comme l'utilisation d'une application de suppression) qui peuvent contextualiser une investigation. L'acquisition de notre image incluait un dump de logcat.

4. Méthodologie d'Analyse

4.1. Vérification et Caractérisation de la Source de Données

La première étape de notre méthodologie a consisté à vérifier et à caractériser la source de données fournie. En nous basant sur la documentation "Android 11 Image Creation Documentation" de Joshua Hickman et les journaux d'acquisition de Magnet ACQUIRE (version 2.30.0.22097), les informations suivantes ont été établies concernant l'appareil d'origine et l'image :

- ***Modèle : Google Pixel 3 (G013A)***
- ***Numéro de Série : 8CEX1N716***
- ***OS : Android 11 (Build RP1A.200720.009, Patch de sécurité 2020-09-05)***
- ***État de Root : Rooté via Magisk***
- ***Chiffrement : File-Based Encryption (FBE)***
- ***Profils Utilisateur : Deux profils configurés (User 1 : thisisdfir@gmail.com, User 2 "Guest" : thisisdfirtwo@gmail.com)***
- ***Passcodes (0731 User1 ; 1234 User2).***

La vérification de l'intégrité de l'archive ont été faites avec les commandes linux md5sum et sha256sum : Les hachages MD5, SHA1 et SHA256 du fichier Android 11 - Pixel 3 - Data.tar ont été comparés aux valeurs fournies dans image_info.txt et se sont avérés concordants, confirmant l'intégrité de la source de données.

4.2. Préparation de l'Environnement d'Analyse et Traitement Initial des Données

Après la vérification de l'intégrité de l'archive « Android 11 - Pixel 3 - Data.tar », celle-ci a été décompressée à l'aide de **winRar** dans un répertoire de travail dédié.

Le contenu extrait, représentant la partition « /data », a ensuite été ajouté comme source de données de type « Fichiers Logiques » (Logical File) dans un nouveau cas créé avec l'outil Autopsy (version 4.22.0). *(Voir Annexe 1 pour des captures d'écran de l'interface d'Autopsy avec la source de données chargée)*.

Une première exploration dans Autopsy a confirmé l'absence de fichiers dans la vue « Fichiers Supprimés » (issus du système de fichiers), ce qui est attendu pour une acquisition logique ne contenant pas l'espace non alloué.

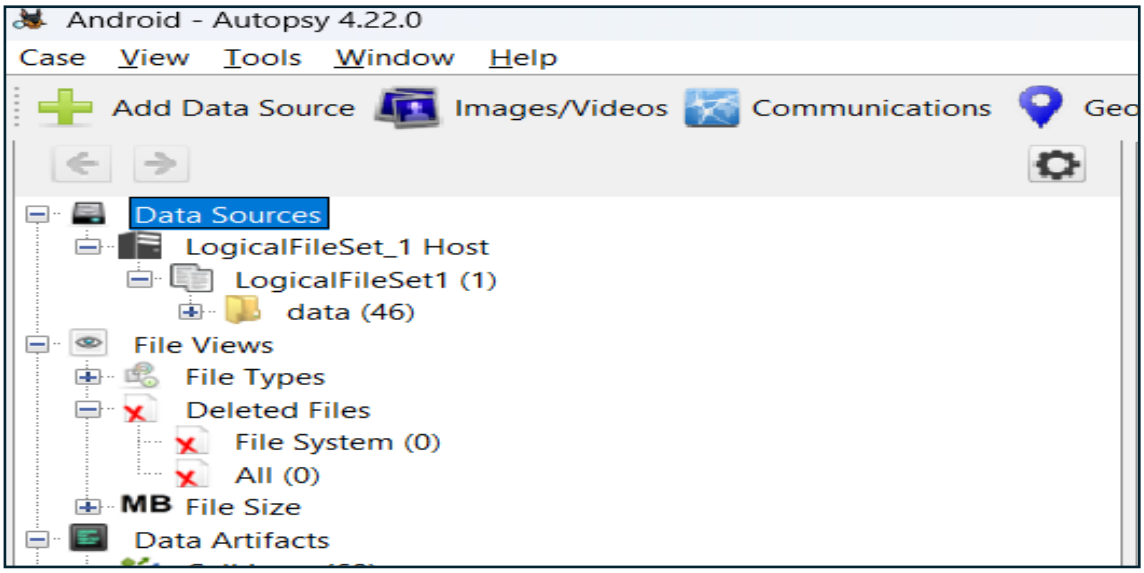


Figure 5: Pas de fichiers supprimés visible

4.3. Analyse Ciblée des Artefacts de Google Photos

Conformément au scénario d'investigation basé sur la documentation de création de l'image, notre analyse s'est concentrée sur les artefacts de l'application Google Photos, localisés dans le répertoire /data/data/com.google.android.apps.photos/databases/.

Les bases de données SQLite « gphotos-1.db » (ainsi que ses fichiers auxiliaires -shm et -wal) et « local_trash.db » ont été identifiées comme potentiellement pertinentes.

Ces bases de données ont été examinées à l'aide de DB Browser for SQLite 3.13.1 pour identifier les tables et les champs relatifs à la gestion des médias et à leur suppression. Une attention particulière a été portée à la table « media » dans « gphotos-1.db » et à la table `local` dans « local_trash.db », en recherchant spécifiquement des colonnes telles que « trash_timestamp » et « is_deleted ». *(Voir annexe 2 pour des captures d'écran).*

4.4. Analyse des Logs Système « logcat »

L'extrait de « logcat » fourni par l'outil d'acquisition Magnet ACQUIRE (contenu dans « activity_log.txt ») a été analysé. Des recherches par mots-clés ***(ex: 'delete', 'remove', 'com.google.android.apps.photos')*** et un filtrage chronologique autour des heures de suppression documentées ont été effectués.

5. Analyse des Résultats et Découvertes

5.1. Analyse des Bases de Données de Google Photos

5.1.1. Examen de local_trash.db

Cette absence de données dans « local_trash.db » pourrait s'expliquer par plusieurs facteurs : un vidage manuel de la corbeille par l'utilisateur avant l'acquisition de l'image, un mécanisme de purge automatique des entrées de cette base de données spécifique, ou encore le fait que les suppressions documentées aient été gérées et journalisées principalement au niveau des bases de données principales de l'application (gphotos-1.db) plutôt que dans cette table « local_trash » dédiée.

5.1.2. Identification des Éléments Mis à la Corbeille dans la Table media de gphotos-1.db

L'analyse de la table media de la base de données gphotos-1.db a permis d'identifier des artefacts cruciaux relatifs à la suppression de fichiers. Plusieurs entrées présentaient des valeurs non nulles dans la colonne trash_timestamp, un indicateur clé de la mise à la corbeille d'un élément par l'application Google Photos.

Plus précisément, trois entrées se distinguent par des trash_timestamp qui, après conversion en format UTC (Coordinated Universal Time), correspondent étroitement aux actions de suppression d'images documentées par Joshua Hickman pour la date du 2020-10-03 :

- **Entrée A (associée à l'autre suppression de 21:02 EDT) :**
    - trash_timestamp : 1601773355000
    - Converti en UTC : **2020-10-04 01:02:35 UTC**
    - Correspondance : Action documentée à 2020-10-03 21:02 EDT (soit 2020-10-04 01:02 UTC).
- **Entrée B (associée à une des suppressions de 21:02 EDT) :**
    - trash_timestamp : 1601773327000
    - Converti en UTC : **2020-10-04 01:02:07 UTC**

- Correspondance : Action documentée à 2020-10-03 21:02 EDT (soit 2020-10-04 01:02 UTC).

- **Entrée C (associée à la suppression de 21:01 EDT) :**
  - trash_timestamp : 1601773261000
  - Converti en UTC : **2020-10-04 01:01:01 UTC**
  - Correspondance : Action documentée à 2020-10-03 21:01 EDT (soit 2020-10-04 01:01 UTC).

Pour ces trois entrées, la colonne is_deleted affichait la valeur '1', corroborant leur statut de suppression au sein de. La colonne « dedup_key » contenait des identifiants uniques similaires à **« ZZn7YFRCNFwdfh5EuqmngNvjYgk »**, qui se sont avérés utiles pour corréler ces enregistrements avec d'autres tables, comme nous le verrons en section 5.1.3. (Voir Annexe 2 pour une capture d'écran illustrant ces entrées dans la table media).

Il est également à noter qu'un grand nombre d'autres entrées dans cette table media présentaient la colonne is_hidden à '1', ce qui suggère des fichiers archivés par l'utilisateur ou des médias internes à l'application, distincts des éléments spécifiquement placés en corbeille. (Voir Annexe 2 pour une capture d'écran illustrant ces entrées dans la table media).

5.1.3. Corrélation avec la Table local_media : Identification des Chemins d'Accès et Analyse des purge_timestamp

Bien que la table media (analysée en section 5.1.2) ait fourni les trash_timestamp confirmant la mise à la corbeille, elle ne contenait pas directement le chemin d'accès local original des fichiers concernés. Pour obtenir cette information, une corrélation a été établie avec la table local_media, présente également dans la base de données gphotos-1.db.

Cette corrélation a été effectuée en utilisant la colonne dedup_key comme clé de jointure entre les enregistrements de la table media (pour les trois éléments mis à la corbeille) et ceux de la table local_media. Cette démarche a permis d'identifier les informations suivantes pour les trois images mises à la corbeille (dont les trash_timestamp ont été présentés en section 5.1.2) :

- **Image 1 (mise à la corbeille le 2020-10-04 01:01:01 UTC) :**
    - Nom du fichier: *.trashed-1604365261-Screenshot_20201003-204952.png*

      Chemin d'accès local:

      /storage/emulated/0/Pictures/Screenshots/.trashed-1604365261-Screenshot_20201003-204952.png
    - purge_timestamp (converti) : **Mardi 3 Novembre 2020 01:01:01 UTC**
- **Image 2 (mise à la corbeille le 2020-10-04 01:02:07 UTC) :**
    - Nom du fichier: *.trashed-1604365327-Screenshot_20201003-205827.png*
    - Chemin d'accès local :

      /storage/emulated/0/Pictures/Screenshots/.trashed-1604365327-Screenshot_20201003-205827.png
    - purge_timestamp (converti) : **Mardi 3 Novembre 2020 01:02:07 UTC**
- **Image 3 (mise à la corbeille le 2020-10-04 01:02:35 UTC) :**
    - Nom du fichier: *.trashed-1604365355-Screenshot_20201003-210010.png*
    - Chemin d'accès local:

      /storage/emulated/0/Pictures/Screenshots/.trashed-1604365355-Screenshot_20201003-210010.png
    - purge_timestamp (converti) : **Mardi 3 Novembre 2020 01:02:35 UTC**

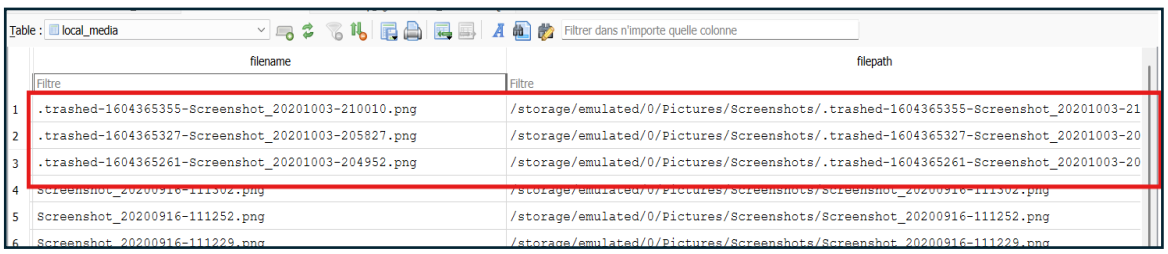
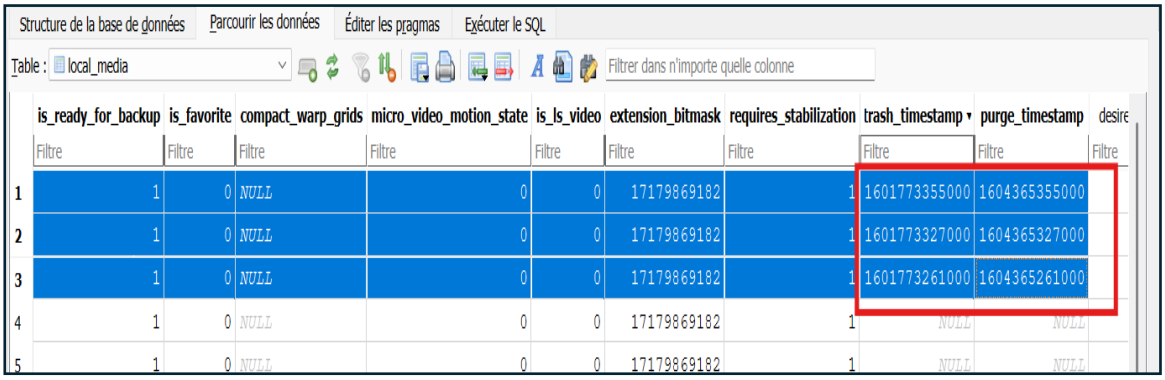


Table : local_media

Filtrer dans n'importe quelle colonne

	filename	filepath
	Filtre	Filtre
1	.trashed-1604365355-Screenshot_20201003-210010.png	/storage/emulated/0/Pictures/Screenshots/.trashed-1604365355-Screenshot_20201003-21
2	.trashed-1604365327-Screenshot_20201003-205827.png	/storage/emulated/0/Pictures/Screenshots/.trashed-1604365327-Screenshot_20201003-20
3	.trashed-1604365261-Screenshot_20201003-204952.png	/storage/emulated/0/Pictures/Screenshots/.trashed-1604365261-Screenshot_20201003-20
4	Screenshot_20200916-111302.png	/storage/emulated/0/Pictures/Screenshots/Screenshot_20200916-111302.png
5	Screenshot_20200916-111252.png	/storage/emulated/0/Pictures/Screenshots/Screenshot_20200916-111252.png
6	Screenshot_20200916-111229.png	/storage/emulated/0/Pictures/Screenshots/Screenshot_20200916-111229.png

Figure 6: Présence de colonnes "filename" et "filepath" dans la table local_media



	is_ready_for_backup	is_favorite	compact_warp_grids	micro_video_motion_state	is_ls_video	extension_bitmask	requires_stabilization	trash_timestamp	purge_timestamp	desire
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	0	NULL	0	0	17179869182	1	1601773355000	1604365355000	
2	1	0	NULL	0	0	17179869182	1	1601773327000	1604365327000	
3	1	0	NULL	0	0	17179869182	1	1601773261000	1604365261000	
4	1	0	NULL	0	0	17179869182	1	NULL	NULL	
5	1	0	NULL	0	0	17179869182	1	NULL	NULL	

Figure 7: Présence de colonne purge_timestamp dans la table local_media

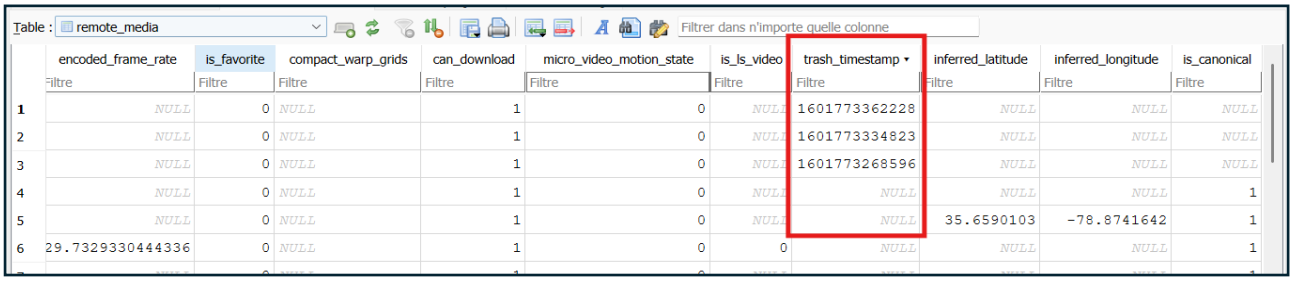
L'analyse des valeurs de la colonne purge_timestamp pour ces trois éléments est particulièrement instructive. Elles indiquent une date de purge prévue environ 30 jours après leur mise à la corbeille. Étant donné que l'acquisition de l'image système a eu lieu le 5 Octobre 2020, ces purge_timestamp (fixés au 3 Novembre 2020) confirment que, au moment de l'acquisition, ces fichiers, bien que mis à la corbeille, n'avaient pas encore atteint leur date de purge définitive et étaient donc toujours considérés comme "récupérables" depuis la corbeille de l'application Google Photos. Cela souligne la persistance des métadonnées relatives aux fichiers supprimés et la fenêtre de temps pendant laquelle des traces peuvent être retrouvées.

5.1.4. Analyse des Artefacts de Suppression dans la Table remote_media

L'investigation a été étendue à la table « remote_media », identifiée principalement dans la base de données gphotos0.db. Cette table est présumée contenir les métadonnées des fichiers synchronisés avec le service cloud de Google Photos.

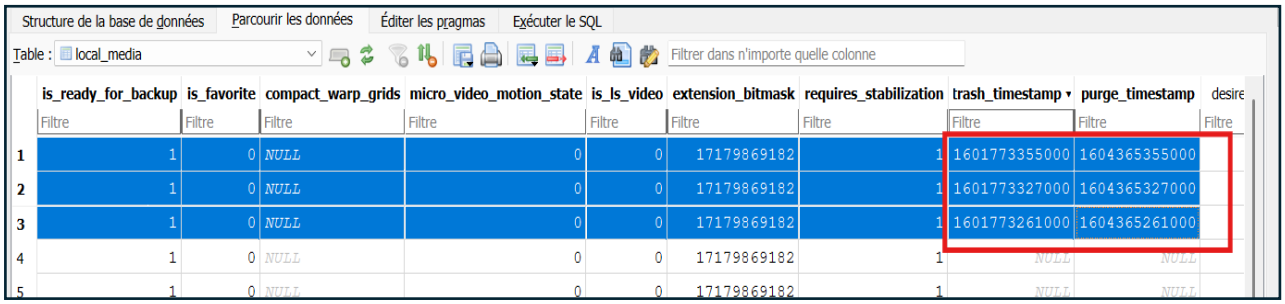
Une découverte significative concerne les entrées correspondant potentiellement aux trois mêmes images précédemment identifiées comme mises à la corbeille localement (section 5.1.2 et 5.1.3). Dans la table remote_media, des enregistrements ont été trouvés avec des noms de fichiers (filename) correspondant aux noms originaux de ces captures d'écran (par exemple, Screenshot_20201003-210010.png), sans le préfixe .trashed-[timestamp]- observé dans la table local_media.

Ces entrées dans « remote_media » présentaient également des valeurs non nulles dans leur colonne trash_timestamp. Cependant, ces timestamps différaient de ceux enregistrés pour les copies locales, étant **systématiquement postérieurs d'environ 7 secondes**, comme détaillé ci-dessous :



	encoded_frame_rate	is_favorite	compact_warp_grids	can_download	micro_video_motion_state	is_ls_video	trash_timestamp	inferred_latitude	inferred_longitude	is_canonical
1	NULL	0	NULL	1	0	NULL	1601773362228	NULL	NULL	NULL
2	NULL	0	NULL	1	0	NULL	1601773334823	NULL	NULL	NULL
3	NULL	0	NULL	1	0	NULL	1601773268596	NULL	NULL	NULL
4	NULL	0	NULL	1	0	NULL	NULL	NULL	NULL	1
5	NULL	0	NULL	1	0	NULL	NULL	35.6590103	-78.8741642	1
6	29.7329330444336	0	NULL	1	0	0	NULL	NULL	NULL	1

Figure 8: Présence de colonne trush_timestamp dans la table remote_media



	is_ready_for_backup	is_favorite	compact_warp_grids	micro_video_motion_state	is_ls_video	extension_bitmask	requires_stabilization	trash_timestamp	purge_timestamp
1	1	0	NULL	0	0	17179869182	1	1601773355000	1604365355000
2	1	0	NULL	0	0	17179869182	1	1601773327000	1604365327000
3	1	0	NULL	0	0	17179869182	1	1601773261000	1604365261000
4	1	0	NULL	0	0	17179869182	1	NULL	NULL
5	1	0	NULL	0	0	17179869182	1	NULL	NULL

Figure 9: Présence de colonne "trush_timestamp" et "purge_timestamp » dans la table local_media

- **Image 1 (Original : Screenshot_20201003-204952.png) :**

trash_timestamp dans remote_media (converti UTC): **GMT**: Sunday 4 October 2020 01:01:08.596

(Rappel : trash_timestamp local était 2020-10-04 01:01:01 UTC)

- **Image 2 (Original : *Screenshot_20201003-205827.png*) :**

trash_timestamp dans remote_media (converti UTC): **GMT**: Sunday 4 October 2020 01:02:14.823

(Rappel : trash_timestamp local était 2020-10-04 01:02:07 UTC)

- **Image 3 (Original : Screenshot_20201003-210010.png):**

trash_timestamp dans remote_media (converti UTC): **GMT**: Sunday 4 October 2020 01:02:42.228

(Rappel : trash_timestamp local était 2020-10-04 01:02:35 UTC)

Ces différences de trash_timestamp entre les artefacts locaux (local_media et media) et les artefacts cloud (remote_media) pour les mêmes fichiers originaux suggèrent que le processus de mise à la corbeille est journalisé distinctement au niveau local et au niveau du service

cloud. Le timestamp dans remote_media, étant constamment postérieur de quelques secondes à l'action locale, pourrait indiquer le moment où la suppression a été effectivement synchronisée ou traitée par les serveurs de Google Photos, après un court délai de propagation.

Cette observation souligne la complexité de l'analyse des applications avec une forte composante cloud. Les actions d'un utilisateur sur son appareil peuvent générer des traces multiples et horodatées différemment à travers les différentes couches de stockage et de synchronisation (local, cloud). Une investigation complète nécessiterait de corréler ces différents jeux de timestamps pour reconstituer l'ensemble du cycle de vie d'un fichier supprimé.

5.2. Analyse des Logs Système (logcat)

L'acquisition de l'image par Magnet ACQUIRE incluait un dump des logs système (logcat) capturés au moment de l'opération. Cet extrait, disponible dans le fichier activity_log.txt, a été examiné dans le but d'identifier d'éventuelles traces corroborant ou contextualisant les actions de suppression d'images.

5.2.1. Méthodologie d'Analyse du logcat

L'analyse de l'extrait de logcat s'est concentrée sur la période entourant les suppressions d'images documentées (2020-10-04, entre 01:01 UTC et 01:03 UTC environ). Les approches suivantes ont été utilisées :

- **Filtrage Chronologique :** Examen des entrées dont les timestamps se situaient dans la fenêtre temporelle d'intérêt.
- **Recherche par Mots-Clés :** Utilisation de mots-clés pertinents tels que "delete", "remove", "trash", "unlink", "Google Photos", "com.google.android.apps.photos", ainsi que les noms de fichiers identifiés (si connus à ce stade de l'analyse des logs).
- **Identification des Processus :** Attention particulière portée aux messages émis par les processus liés à Google Photos, au MediaStore d'Android, ou à des services système de gestion de fichiers.

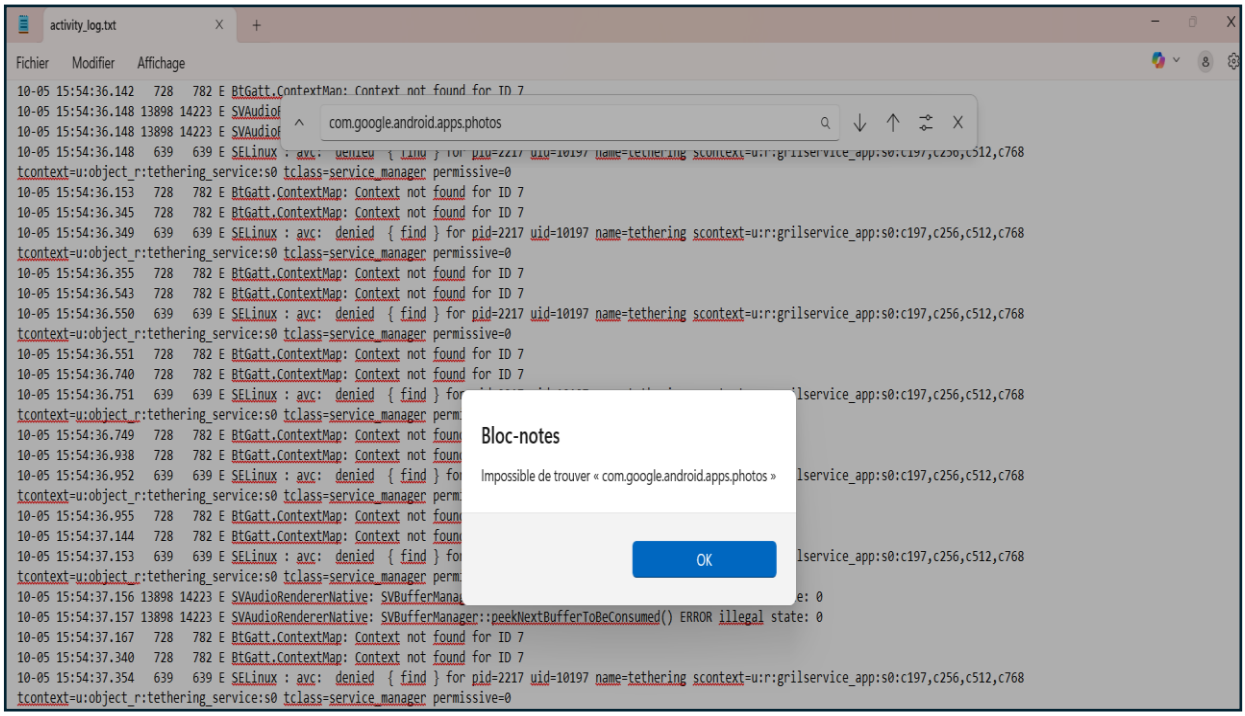


Figure 11 : Recherche de mot de clé

5.2.2. Constatations Issues de l'Analyse du logcat

L'extrait de logcat fourni par l'outil d'acquisition contenait principalement des messages d'erreur ou d'information relatifs à d'autres composants système ou applications (par exemple, des erreurs BtGatt.ContextMap, SVAudioRendererNative, ou des dénis SELinux, comme observé dans activity_log.txt).

Malgré une recherche ciblée, **aucune entrée directement et explicitement identifiable comme une journalisation de la suppression des trois images spécifiques (par exemple, un log indiquant "Fichier X supprimé par Google Photos") n'a été trouvée dans l'extrait de logcat disponible pour la période concernée.**

Plusieurs facteurs peuvent expliquer cette absence d'entrées directes :

- **Niveau de verbosité des logs :** L'application Google Photos ou le système Android peuvent ne pas journaliser les opérations de suppression de fichiers utilisateur à un niveau de détail suffisant dans les buffers de logcat standards.
- **Nature circulaire des buffers logcat :** Les logs pertinents, s'ils ont existé, ont pu être écrasés par des messages plus récents avant le dump effectué par Magnet ACQUIRE, bien que le dump ait été réalisé peu de temps après les actions documentées.

- **Gestion interne par l'application :** Les opérations de mise à la corbeille peuvent être gérées de manière interne par l'application Google Photos sans générer de messages logcat explicites pour chaque fichier.

Bien que cette analyse du logcat n'ait pas fourni de confirmation directe des suppressions, elle constitue une étape nécessaire de la méthodologie d'investigation pour explorer toutes les sources de données disponibles.

6. Étude de Cas Pratique : Récupération par File Carving

6.1. Justification et Objectif de la Simulation

L'analyse principale de ce rapport s'est concentrée sur l'examen des artefacts présents au sein d'une archive logique (.tar) de la partition /data d'un appareil Android. Comme discuté précédemment (Section 2.2 et 4.2), la nature de cette acquisition logique, bien que permettant une analyse détaillée des fichiers alloués et des bases de données applicatives, ne permet pas d'accéder à l'espace non alloué du stockage. Or, c'est précisément dans cet espace non alloué que résident souvent les données résiduelles des fichiers supprimés qui ne sont plus référencés par le système de fichiers ou les applications (par exemple, après un vidage de la corbeille ou une suppression plus "profonde").

Afin d'aborder l'aspect "récupération de données effacées" du sujet de manière plus complète et de démontrer une technique fondamentale utilisée lorsque les métadonnées sont absentes ou insuffisantes, une simulation contrôlée de récupération de fichiers par "file carving" a été réalisée. Cette technique, qui s'appuie sur la reconnaissance des signatures (en-têtes et pieds de page) spécifiques aux différents types de fichiers, est particulièrement pertinente lors de l'analyse d'images physiques complètes d'un support de stockage.

L'objectif principal de cette simulation est donc double :

1. Illustrer le principe et la méthodologie du file carving pour la récupération de fichiers supprimés.
2. Démontrer l'efficacité potentielle et les limites de cette technique dans un environnement contrôlé, en complément de l'analyse des artefacts applicatifs réalisée sur l'image Android principale.

Cette étude de cas pratique vise à enrichir la compréhension des différentes approches disponibles pour un investigateur en digital forensics face à des données effacées.

6.2. Protocole de Préparation de l'Image de Test (image_test_carving.dd)

6.2.1. Création d'une image disque brute

Un fichier image disque, nommé image_test_carving.dd, d'une taille de 100 MiB (environ 105 MB) a été créé à l'aide de la commande dd. Ce fichier a été initialisé avec des zéros pour simuler un espace de stockage non structuré. La commande exacte utilisée dans le terminal

*Figure 12: **Création du fichier image image_test_carving.dd avec la commande dd.***

6.2.2. Formatage de l'image disque

L'image disque image_test_carving.dd a ensuite été attachée à un périphérique "loopback" (/dev/loop0) au sein de WSL. Elle a été formatée avec le système de fichiers FAT32 à l'aide de la commande **mkfs.vfat** (du paquet **dosfstools**), la rendant ainsi apte à stocker des fichiers :

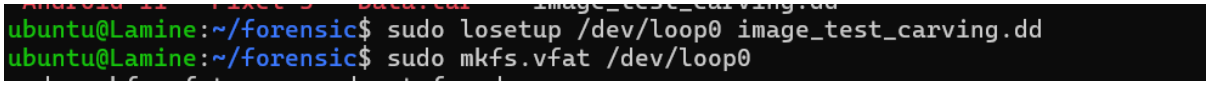


Figure 13: Formatage de l'image disque

6.2.3. Montage et Peuplement du volume

Le volume FAT32 a été monté sur le point de montage ~/test_mount en spécifiant les options uid et gid de l'utilisateur ubuntu pour assurer les permissions d'écriture :

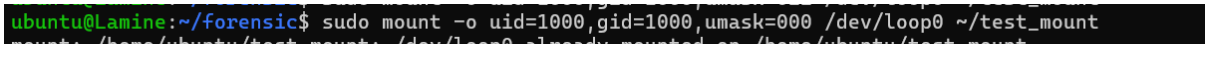


Figure 14 : Montage du volume

Plusieurs fichiers exemples, incluant des images JPEG, PNG, et un document PDF, extraits du /data/ , ont été copiés sur ce volume.

```
ubuntu@Lamine:~/forensic$ ls -l ~/test_mount
total 59128
-rwxr-xr-x 1 ubuntu ubuntu        0 Jun  2 01:13  1601776206974.mp4
-rwxr-xr-x 1 ubuntu ubuntu        0 Jun  2 01:15  17069-1601776206974.mp4
-rwxr-xr-x 1 ubuntu ubuntu    50831 Jun  2 01:12  350x350bb.jpg
-rwxr-xr-x 1 ubuntu ubuntu        0 Jun  2 01:15  38114-AUDIO.mp4
-rwxr-xr-x 1 ubuntu ubuntu        0 Jun  2 01:15  38121-VID.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1313337 Jun  2 01:15  49815-d3jzo1isir3lryogrxxz.mp4
-rwxr-xr-x 1 ubuntu ubuntu   461296 Jun  2 01:15  49824-86ad19b81bbce71450e58154943665378e3e381e_v23_phone.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1623021 Jun  2 01:15  49995-0f0f9628a054c2c35ccf1cd3f0e51205.mp4
-rwxr-xr-x 1 ubuntu ubuntu   970484 Jun  2 01:15  49997-23f3848faf0c2f3af6379e407508b928.mp4
-rwxr-xr-x 1 ubuntu ubuntu   501292 Jun  2 01:15  49998-2696ab606874037a1d1ed7cb880d28a0.mp4
-rwxr-xr-x 1 ubuntu ubuntu  3522948 Jun  2 01:15  49999-2e5cc593d66bd0ef015f71f9a5fd07fe.mp4
-rwxr-xr-x 1 ubuntu ubuntu   247597 Jun  2 01:15  50000-4324d44b789ff12c89bdbc6b36dd3f25.mp4
-rwxr-xr-x 1 ubuntu ubuntu  4152163 Jun  2 01:15  50001-455cf28b1f2901432866a732b6bac93a.mp4
-rwxr-xr-x 1 ubuntu ubuntu   240001 Jun  2 01:15  50002-5556a578248cffb05b9f42dc6bea6a55.mp4
-rwxr-xr-x 1 ubuntu ubuntu  5386075 Jun  2 01:15  50003-573d7c23ebc66d4beff4f3773489c0cd.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1467761 Jun  2 01:15  50004-5f27955cf38891d96786bb16fa9fe677.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1530735 Jun  2 01:15  50005-656df1ed312589146947b3b3c976e4e1.mp4
-rwxr-xr-x 1 ubuntu ubuntu  3135458 Jun  2 01:15  50006-658038754258b00aac436fc46c883559.mp4
-rwxr-xr-x 1 ubuntu ubuntu     8389 Jun  2 01:15  50008-7b149ca8a74237dce670c8155c30efa6.mp4
-rwxr-xr-x 1 ubuntu ubuntu   920593 Jun  2 01:15  50009-880df22142acae43ec57e90e09689558.mp4
-rwxr-xr-x 1 ubuntu ubuntu  6206799 Jun  2 01:15  50011-8e3217741df438f89c08d24bbbe196c3.mp4
-rwxr-xr-x 1 ubuntu ubuntu  8560428 Jun  2 01:15  50012-d4da6754a7bb6e52a3f40b5708b394de.mp4
-rwxr-xr-x 1 ubuntu ubuntu   207472 Jun  2 01:15  50013-d83279f8c530b51928eaa8e4c5e6a4b6.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1840341 Jun  2 01:15  50014-d8adbcc1434e4c5e89f841b09056bc06.mp4
-rwxr-xr-x 1 ubuntu ubuntu   250041 Jun  2 01:15  50015-e60f18aa8ae5d40a1ccbb3ed138ca300.mp4
-rwxr-xr-x 1 ubuntu ubuntu   565306 Jun  2 01:15  50016-e69119260c4246b95b804ba6d32b90fe.mp4
-rwxr-xr-x 1 ubuntu ubuntu  1984699 Jun  2 01:15  50017-e988afcdb12bf8026a2e949a20b31a32.mp4
-rwxr-xr-x 1 ubuntu ubuntu    53892 Jun  2 01:15  50018-fdb5f4e3be2303972b1208bf855c40b6.mp4
-rwxr-xr-x 1 ubuntu ubuntu   520925 Jun  2 01:15  51579-welcome_screen_video4.mp4
-rwxr-xr-x 1 ubuntu ubuntu    23959 Jun  2 01:15  53266-400050000226_30815.mp4
-rwxr-xr-x 1 ubuntu ubuntu    36580 Jun  2 01:15  53267-400050200562_26829.mp4
-rwxr-xr-x 1 ubuntu ubuntu    35871 Jun  2 01:15  53268-400050700960_24639.mp4
-rwxr-xr-x 1 ubuntu ubuntu    67379 Jun  2 01:15  53269-400061600179_25910.mp4
-rwxr-xr-x 1 ubuntu ubuntu    55555 Jun  2 01:15  53270-400062100389_29272.mp4
```

Figure 15 : Preuve du peuplement du volume

6.2.4. Suppression des fichiers exemples

Plusieurs d'entre les fichiers copiés a ensuite été supprimé du volume ~/test_mount à l'aide de la commande rm.

```
ubuntu@Lamine:~/forensic$ rm ~/test_mount/*5*
ubuntu@Lamine:~/forensic$
ubuntu@Lamine:~/forensic$ ls -l ~/test_mount
total 1924
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:13  1601776206974.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  17069-1601776206974.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  38114-AUDIO.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  38121-VID.mp4
-rwxr-xr-x 1 ubuntu ubuntu  501292 Jun  2 01:15  49998-2696ab606874037a1d1ed7cb880d28a0.mp4
-rwxr-xr-x 1 ubuntu ubuntu  101470 Jun  2 01:12  600x600bb.jpg
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  72724-1601776206974.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  92824-AUDIO.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:15  92831-VID.mp4
-rwxr-xr-x 1 ubuntu ubuntu       0 Jun  2 01:13  AUDIO.mp4
-rwxr-xr-x 1 ubuntu ubuntu   49482 Jun  2 01:11 'Google Fi-2020-07-17.pdf'
-rwxr-xr-x 1 ubuntu ubuntu 1313337 Jun  2 01:13  d3jzo1isir3lryogrxxz.mp4
```

Figure 16: Suppression des fichiers exemples

6.2.5. Démontage et Détachement

Enfin, le volume a été démonté et le périphérique loopback détaché pour finaliser l'image de test :

```
ubuntu@Lamine:~/forensic$ sudo umount ~/test_mount
ubuntu@Lamine:~/forensic$ sudo losetup -d /dev/loop0
ubuntu@Lamine:~/forensic$
```

Figure 17: Démontage et Détachement

Le fichier image_test_carving.dd était alors prêt pour l'analyse par file carving

6.3. Application de l'Outil de Carving : Autopsy

L'outil d'analyse forensique Autopsy (version 4.22.0, comme visible sur la capture d'écran) a été utilisé pour effectuer le file carving sur l'image de test image_test_carving.dd. L'image disque brute a été ajoutée comme une nouvelle source de données de type "Disk Image" au sein d'un cas Autopsy. Lors de la phase d'ingestion des données, les modules d'analyse pertinents ont été activés, incluant spécifiquement le module de file carving , « PhotoRec Carver ». La configuration du module de carving a été laissée par défaut pour rechercher les types de fichiers courants.

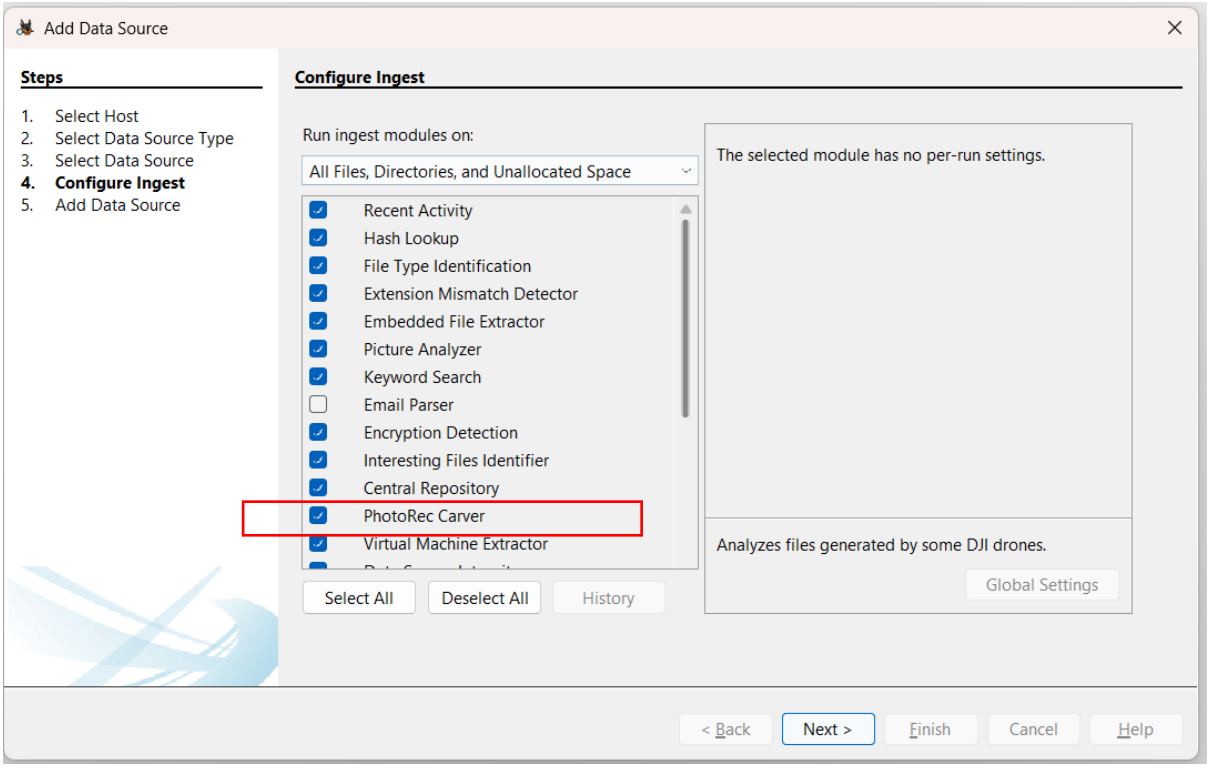


Figure 18: Module carving d'Autopsy

6.4. Résultats de la Récupération par Carving

L'exécution des modules d'ingestion d'Autopsy, et en particulier du module de carving, a permis de récupérer un ensemble de fichiers depuis l'espace non alloué de l'image image_test_carving.dd. Ces fichiers ont été principalement localisés dans la section $CarvedFiles de l'arborescence des sources de données d'Autopsy (voir Figure ci-dessous, illustrant l'interface d'Autopsy avec les fichiers carvés).

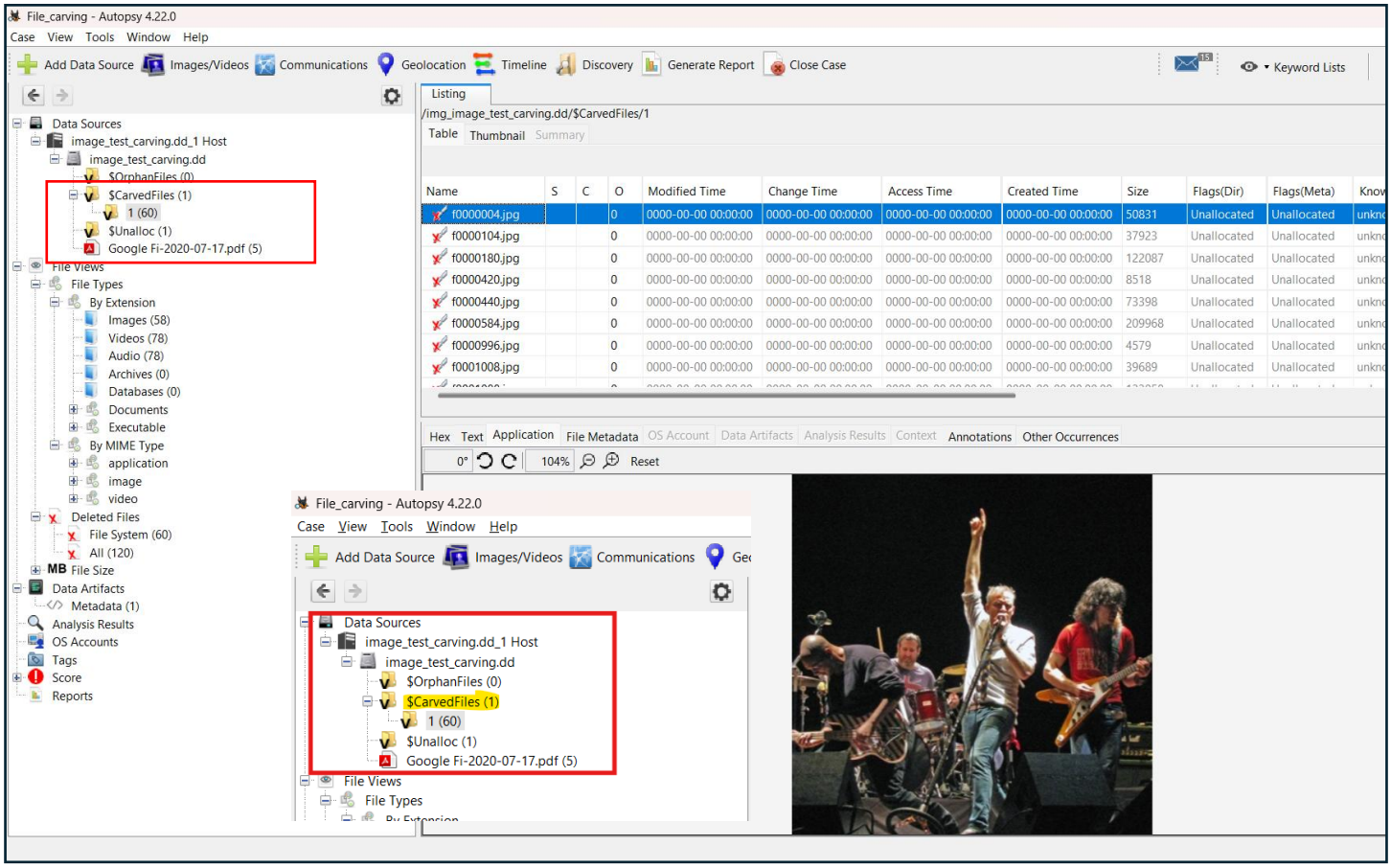


Figure 19 : Interface d'Autopsy avec les fichiers carvés

L'analyse des fichiers récupérés a révélé :

- Images JPEG : Un total de 26 JPG ont été identifiés par le module de carving. Après examen visuel, tous ces fichiers correspondaient aux images JPG originales qui avaient été supprimées et étaient intactes et visualisables (comme illustré par la prévisualisation de f0000004.jpg en Figure 12).

Fichiers mp4 : des fichiers mp4 ont été également carvés.

Les fichiers récupérés par carving portaient des noms génériques assignés par l'outil (ex: f0000004.jpg) et leurs timestamps de système de fichiers (Modification, Accès, Création) n'ont pas pu être restaurés, affichant des valeurs nulles, ce qui est typique de cette technique de récupération.

6.5. Analyse des Résultats et Enseignements de la Simulation

Cette simulation pratique démontre de manière concluante l'efficacité du file carving pour récupérer des données supprimées lorsque les références du système de fichiers ont disparu. Dans notre scénario contrôlé sur une image FAT32, l'outil Autopsy a réussi à identifier et à extraire plusieurs des fichiers images et documents originaux.

Les principaux enseignements de cette démonstration sont :

- **Efficacité du carving sur l'espace non alloué :** La technique a permis de retrouver des fichiers dont les métadonnées FAT32 avaient été effacées lors de la suppression.
- **Perte d'informations contextuelles :** Comme attendu, les noms de fichiers originaux et les timestamps MAC du système de fichiers n'ont pas été récupérés avec les fichiers carvés. Seules les métadonnées internes aux fichiers (comme les données EXIF pour les JPEG, si présentes et récupérées) pourraient fournir des informations chronologiques.

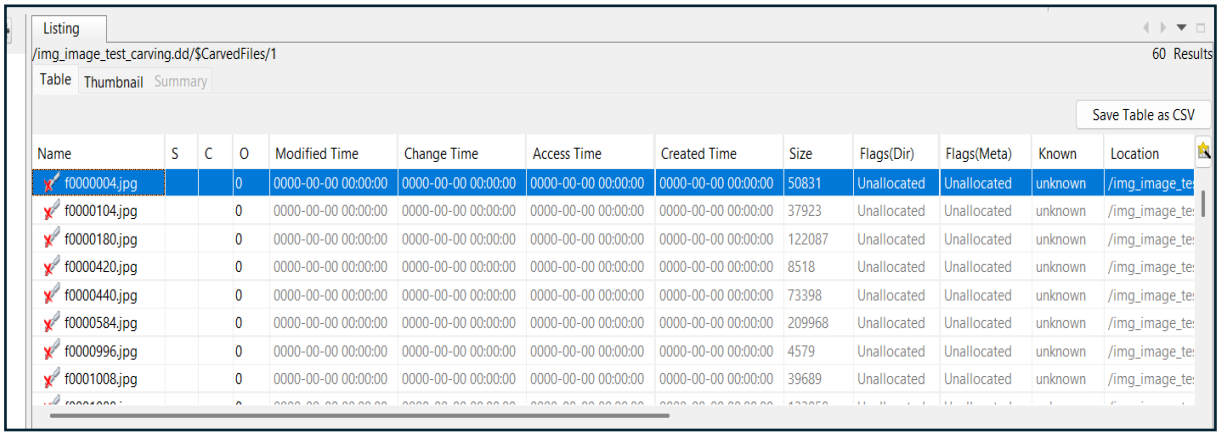


Figure 20 : Perte d'informations contextuelles

- **Importance du type de fichier :** Le succès du carving dépend de la présence de signatures claires et de la non-fragmentation.

- **Pertinence pour les investigations Android :** Bien que cette simulation ait été réalisée sur une image FAT32 simple, le principe du file carving est directement applicable à l'analyse d'images physiques de partitions Android (EXT4, F2FS). Si une image physique complète d'un appareil Android était disponible, le carving serait une méthode essentielle pour tenter de récupérer des fichiers (images, documents, bases de données d'applications, etc.) après leur suppression définitive de la corbeille des applications ou du système.

Cette étude de cas souligne donc l'importance du file carving comme outil complémentaire à l'analyse des artefacts applicatifs et système en forensique mobile.

7. Discussion des Résultats et de la Démarche d'Investigation

L'investigation menée sur l'image Android de test a permis de mettre en lumière plusieurs aspects cruciaux de l'analyse forensique des données supprimées et de la récupération potentielle de celles-ci. Cette section vise à discuter des principales découvertes, des défis rencontrés, et des enseignements tirés.

7.1. Persistance des Métadonnées de Suppression dans les Applications

L'analyse des bases de données de l'application Google Photos (gphotos-1.db et local_media) a été particulièrement instructive. Bien que la base de données local_trash.db ait été trouvée vide, l'identification de trash_timestamp précis dans la table media a permis de corroborer les actions de suppression d'images documentées par le créateur de l'image. De plus, la découverte des purge_timestamp correspondants dans la table local_media a indiqué une politique de rétention de la corbeille d'environ 30 jours, signifiant que les fichiers étaient encore considérés comme "récupérables" depuis l'application au moment de l'acquisition. Ces éléments soulignent la richesse des métadonnées conservées par les applications, même après une action de suppression initiée par l'utilisateur. L'observation de trash_timestamp distincts dans la table remote_media de gphotos0.db pour les mêmes fichiers originaux suggère également la complexité de la journalisation et de la synchronisation des statuts de suppression entre l'appareil local et les services cloud.

7.2. Le File Carving comme Technique Complémentaire Essentielle

La simulation de récupération par file carving, réalisée sur une image disque dédiée (image_test_carving.dd) à l'aide d'Autopsy, a démontré l'efficacité de cette technique pour récupérer des fichiers (JPG, MP4) en se basant uniquement sur leurs signatures, en l'absence de métadonnées du système de fichiers. Bien que cette méthode entraîne la perte des noms de fichiers originaux et des timestamps MAC, elle reste indispensable lorsque les artefacts applicatifs sont absents, que la corbeille a été vidée, ou que l'on travaille sur l'espace non alloué d'une image physique complète. Elle illustre une voie de récupération des données brutes lorsque les traces logiques ont disparu.

7.3. Défis et Limites de l'Investigation sur le Dataset Fourni

Plusieurs défis ont caractérisé cette investigation :

- **Nature de l'Acquisition Logique :** L'utilisation d'une archive .tar de la partition /data a limité la capacité à effectuer des opérations de carving directement sur l'image Android principale, car l'espace non alloué n'était pas inclus. Cela a orienté l'analyse vers les artefacts applicatifs.
- **Analyse du logcat :** L'extrait de logcat fourni, bien qu'analysé méthodiquement, n'a pas révélé d'entrées directement exploitables pour confirmer ou contextualiser les suppressions d'images spécifiques, soulignant la volatilité ou le manque de verbosité des logs pour certaines actions utilisateur.
- **Chiffrement Basé sur les Fichiers (FBE) :** Bien que l'appareil ait été déverrouillé lors de l'acquisition, la présence du FBE sur Android 11 est un facteur complexifiant la récupération de données si les fichiers avaient été définitivement supprimés et leurs clés de chiffrement perdues ou l'espace réécrit.
- **Complexité des Artefacts Applicatifs :** La dispersion des informations pertinentes (chemins, timestamps de suppression, timestamps de purge) entre différentes tables (media, local_media) et potentiellement différentes bases de données (gphotos-1.db, gphotos0.db) au sein d'une même application souligne la nécessité d'une analyse approfondie et de la compréhension des relations entre les données.

7.4. Importance de la Documentation Contextuelle

La documentation fournie par Joshua Hickman concernant la création de l'image de test, ainsi que les journaux d'acquisition de Magnet ACQUIRE, ont été d'une valeur inestimable. Ces informations ont permis de valider les hypothèses, de cibler l'analyse (notamment les suppressions dans Google Photos), de comprendre la configuration de l'appareil (root, FBE), et d'interpréter correctement les timestamps en tenant compte des fuseaux horaires. Sans ce contexte, l'investigation aurait été significativement plus ardue et les conclusions moins assurées.

7.5. Enseignements Tirés

Cette étude de cas a renforcé plusieurs principes clés de la forensique mobile : la persistance des métadonnées au sein des applications, la complémentarité des techniques d'analyse d'artefacts et de file carving, l'impact crucial du type d'acquisition sur les possibilités d'investigation, et l'importance fondamentale d'une documentation et d'un contexte clairs pour mener une analyse rigoureuse.

8. Conclusions

Cette investigation forensique, axée sur la récupération et l'analyse de données supprimées sur un environnement Android 11 (basée sur l'image de test "Android 11 - Pixel 3 - Data.tar"), a permis de mettre en évidence plusieurs aspects significatifs et de répondre aux objectifs initiaux.

Premièrement, l'analyse ciblée des artefacts de l'application Google Photos a démontré **la persistance notable des métadonnées relatives aux fichiers supprimés.** Bien que la base de données dédiée à la corbeille locale (local_trash.db) se soit révélée vide, l'examen des bases de données principales de l'application (notamment gphotos-1.db) a permis d'identifier des trash_timestamp correspondant précisément aux suppressions d'images documentées. De plus, la découverte de purge_timestamp associés dans la table local_media a indiqué une politique de rétention d'environ 30 jours, confirmant que les fichiers étaient toujours considérés comme "récupérables" depuis la corbeille de l'application au moment de l'acquisition de l'image. Ces découvertes soulignent que même après une action de suppression par l'utilisateur, des traces exploitables et des informations chronologiques

peuvent subsister au sein des structures de données des applications. L'analyse a également mis en lumière la complexité introduite par la synchronisation cloud, avec des enregistrements distincts et des timestamps différents pour les mêmes éléments dans les artefacts locaux et distants (remote_media).

Deuxièmement, **la simulation de récupération de fichiers par file carving** a confirmé l'efficacité de cette technique pour extraire des données brutes (images JPEG, PNG, et documents PDF) depuis un espace où les métadonnées du système de fichiers ont été supprimées. Bien que cette méthode entraîne la perte d'informations contextuelles telles que les noms de fichiers originaux et les timestamps MAC, elle demeure une approche indispensable pour la récupération de données lorsque les artefacts applicatifs sont absents ou que les fichiers ont été définitivement retirés de la corbeille et que l'espace n'a pas encore été réécrit, à condition de disposer d'une image physique du support.

Troisièmement, l'investigation a mis en exergue **les défis inhérents à l'analyse d'une acquisition logique** (archive .tar), qui limite l'accès à l'espace non alloué, et a rappelé l'impact potentiel du Chiffrement Basé sur les Fichiers (FBE) sur la récupérabilité des données. L'analyse de l'extrait de logcat fourni, bien que méthodiquement conduite, n'a pas apporté d'éléments directement liés aux suppressions ciblées, illustrant les limites de cette source pour certaines actions spécifiques.

En conclusion, cette étude a réaffirmé l'importance d'une approche multi-facettes en forensique Android, combinant l'analyse approfondie des artefacts applicatifs pour reconstituer les actions et la chronologie, avec la maîtrise des techniques de récupération de bas niveau comme le file carving. Elle a également souligné le rôle crucial d'une documentation contextuelle détaillée (telle que celle fournie par Joshua Hickman et Magnet ACQUIRE) pour guider et valider les investigations numériques.

9. Perspectives et Recommandations

Cette investigation, bien que ciblée sur des aspects spécifiques de la récupération de données supprimées, ouvre plusieurs perspectives pour des analyses plus approfondies et permet de formuler des recommandations pour la pratique forensique sur Android.

9.1. Perspectives d'Analyses Complémentaires

- **Corrélation Contextuelle des Données Récupérées :** Les images identifiées comme ayant été mises à la corbeille (et dont la récupération a été simulée par carving) pourraient faire l'objet d'investigations contextuelles plus poussées. Par exemple, il serait pertinent de rechercher des communications (messages SMS/MMS, emails, messages d'applications de messagerie instantanée) ou d'autres artefacts (entrées de calendrier, notes) qui pourraient être liés à ces images spécifiques. L'analyse chronologique autour des dates de création ou de modification présumées de ces images, ou autour de leur date de suppression, pourrait révéler des échanges ou des activités en lien avec leur contenu ou leur dissimulation. [Ici, tu peux ajouter ta remarque : "Par exemple, un message a été identifié où un utilisateur demande l'envoi d'une photo ; une investigation plus approfondie pourrait tenter de déterminer si l'une des images supprimées correspond à cette requête, bien que cette piste n'ait pas été explorée en détail dans le cadre de ce rapport en raison de sa complexité et des limites de temps."].
- **Analyse Approfondie des Bases de Données Cloud (gphotos0.db) :** L'exploration initiale de la table remote_media a suggéré des actions de suppression distinctes pour les copies cloud des images. Une analyse plus détaillée de gphotos0.db et des mécanismes de synchronisation de Google Photos pourrait apporter un éclairage sur le cycle de vie complet des fichiers entre l'appareil et les serveurs distants.
- **Exploration d'Autres Artefacts de Suppression :** Au-delà de Google Photos, d'autres applications (gestionnaires de fichiers, applications de stockage tierces, applications de communication avec fonctionnalités de suppression) pourraient laisser des traces distinctes. Une analyse exhaustive de l'image .tar pourrait cibler ces applications.
- **Analyse d'une Image Physique Complète :** La réalisation d'une acquisition physique de l'appareil Pixel 3 (si disponible) permettrait d'appliquer les techniques de file carving sur l'ensemble de la partition /data, y compris l'espace non alloué, et de comparer les résultats avec ceux obtenus par l'analyse des artefacts applicatifs et la simulation. Cela permettrait également d'examiner d'autres partitions système.
- **Impact du Chiffrement FBE sur la Récupérabilité :** Des tests spécifiques pourraient être menés pour évaluer plus concrètement l'impact du FBE sur la possibilité de

récupérer des données après différents scénarios de suppression et de réécriture de l'espace sur un appareil similaire.

9.2. Recommandations pour les Investigateurs

Sur la base de cette étude, les recommandations suivantes peuvent être formulées pour les investigations forensiques sur des appareils Android :

- **Privilégier l'Acquisition Physique :** Lorsque cela est techniquement et légalement possible, une acquisition physique (bit-à-bit) du support de stockage devrait être privilégiée pour maximiser les chances de récupération de données supprimées via des techniques comme le file carving.
- **Analyse Approfondie des Artefacts Applicatifs :** Ne pas se limiter aux mécanismes de suppression du système de fichiers.
- Les applications, en particulier celles avec des fonctionnalités cloud ou de gestion de médias, conservent souvent des métadonnées riches (timestamps, statuts de suppression, références à la corbeille) qui peuvent être cruciales.
- **Corrélation et Chronologie Rigoureuses :** Collecter, normaliser (en UTC) et corréler les timestamps provenant de multiples sources (système de fichiers, bases de données applicatives, logs système, documentation externe) est essentiel pour reconstituer une chronologie fiable des événements.
- **Comprendre les Limites des Outils et des Données :** Être conscient des limitations des types d'acquisition (logique vs. physique), de l'impact du chiffrement, de la volatilité des logs, et des spécificités de chaque application analysée.
- **Documentation Exhaustive :** Documenter méticuleusement chaque étape de l'investigation, les outils utilisés, les observations (positives comme négatives), et les interprétations est fondamental pour la validité et la reproductibilité de l'analyse.
- **Veille Technologique Continue :** Les systèmes d'exploitation Android, les applications et les techniques forensiques évoluent rapidement. Une veille continue est nécessaire pour maintenir ses compétences à jour.

Annexes

Annexes 1 : Création du cas sur Autopsy

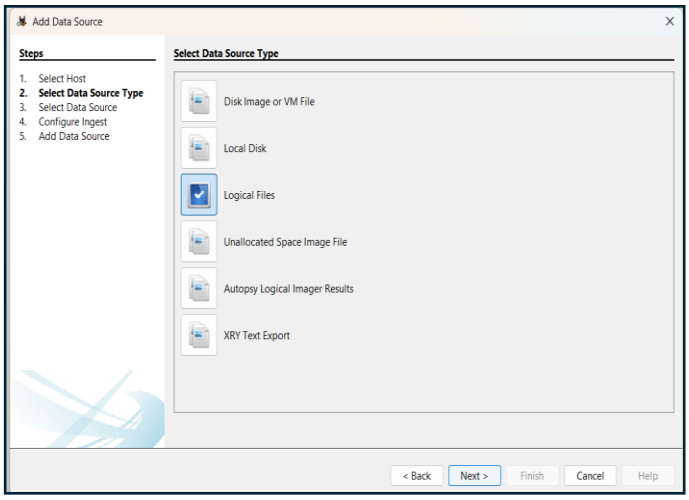


Figure 20 : Choix du type de data source21

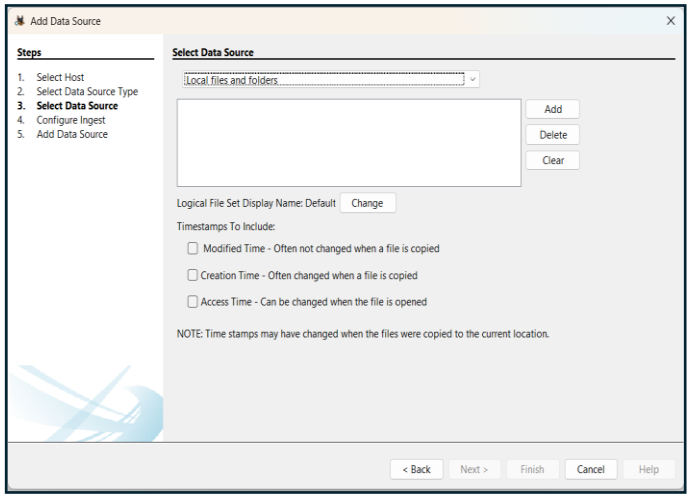


Figure 21 : Interface pour Ajout du data source22

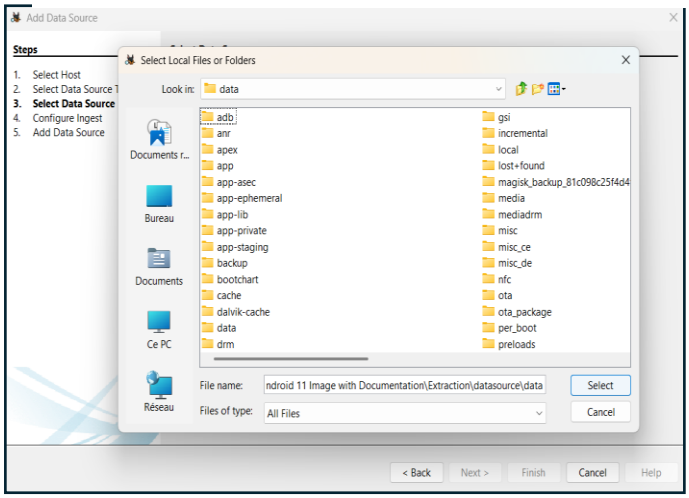


Figure 22 : Ajout du data source23

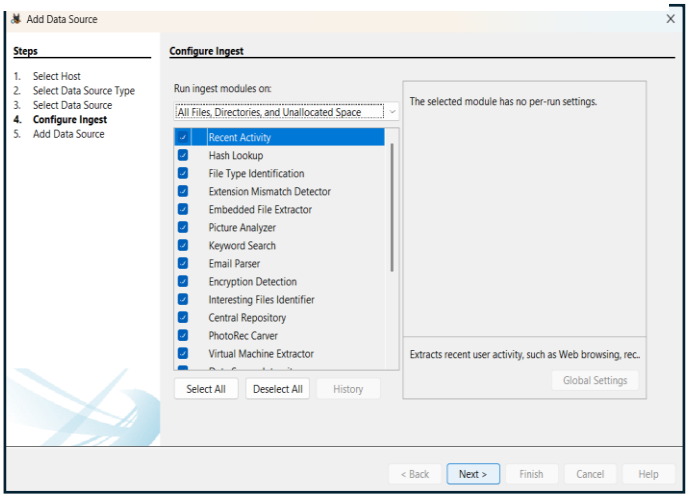


Figure 23 : Sélection des modules25

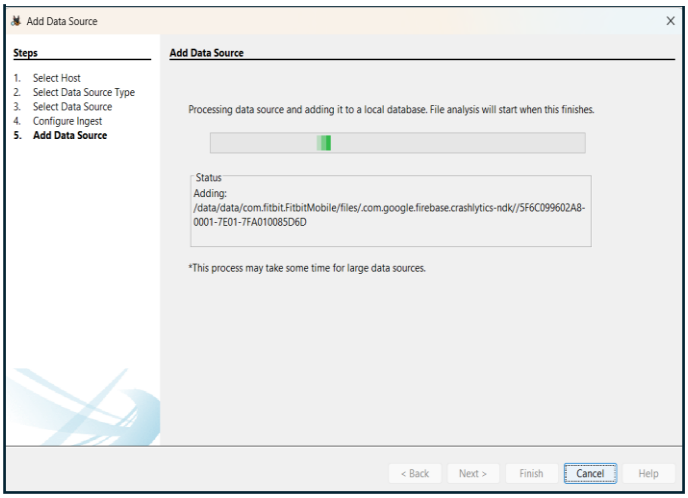


Figure 23 : Chargement du data source dans Autopsy

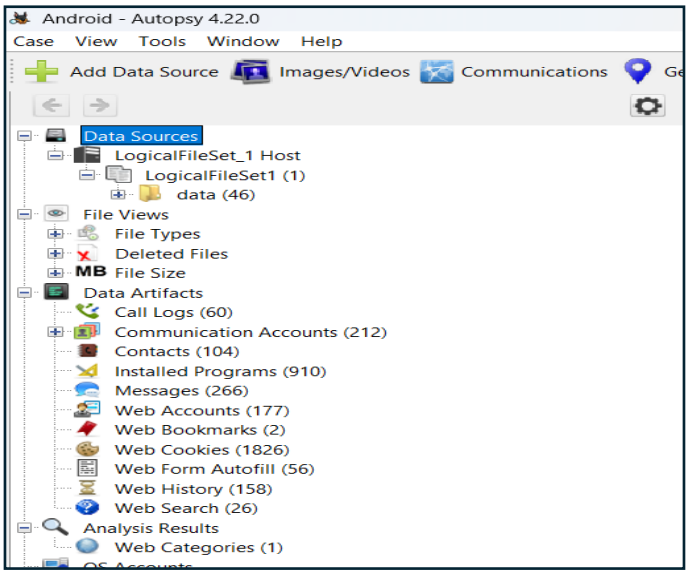
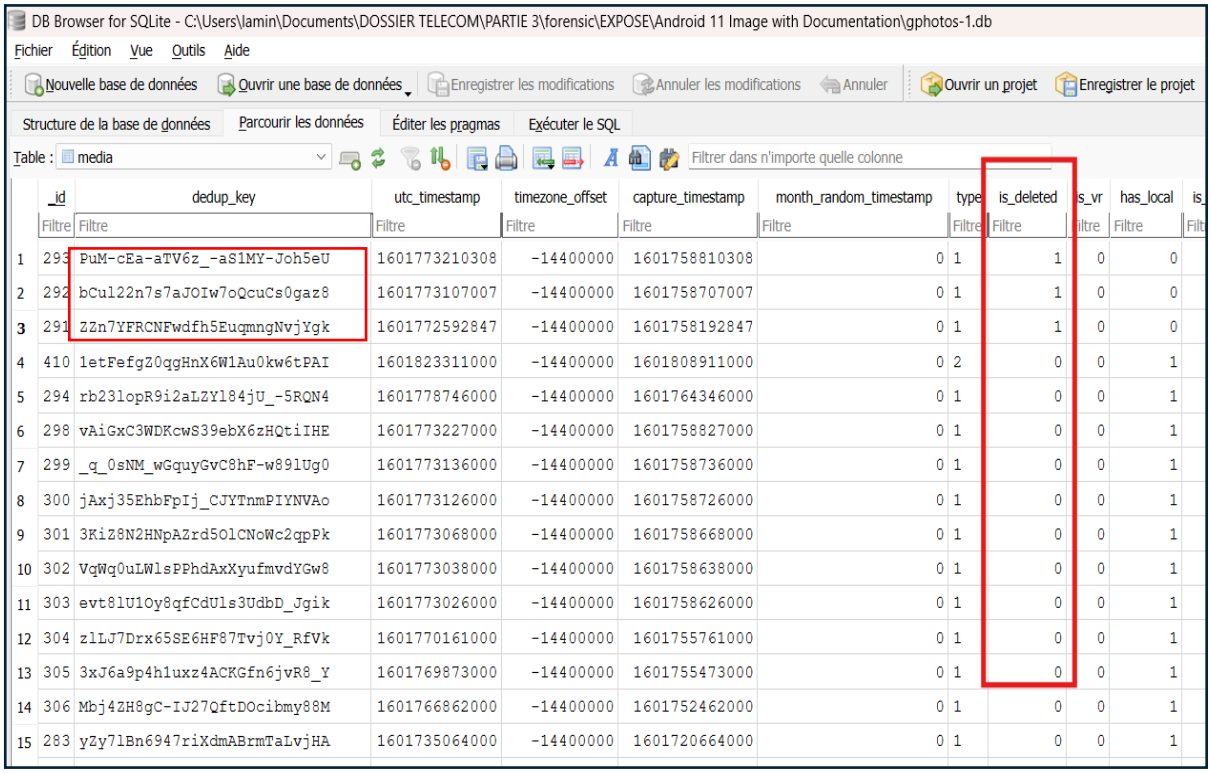


Figure 24 : Data source chargée avec succès dans Autopsy24

Annexes 2 : Création du cas sur Autopsy



DB Browser for SQLite - C:\Users\lamin\Documents\DOSSIER TELECOM\PARTIE 3\forensic\EXPOSE\Android 11 Image with Documentation\gphotos-1.db

Fichier Édition Vue Outils Aide

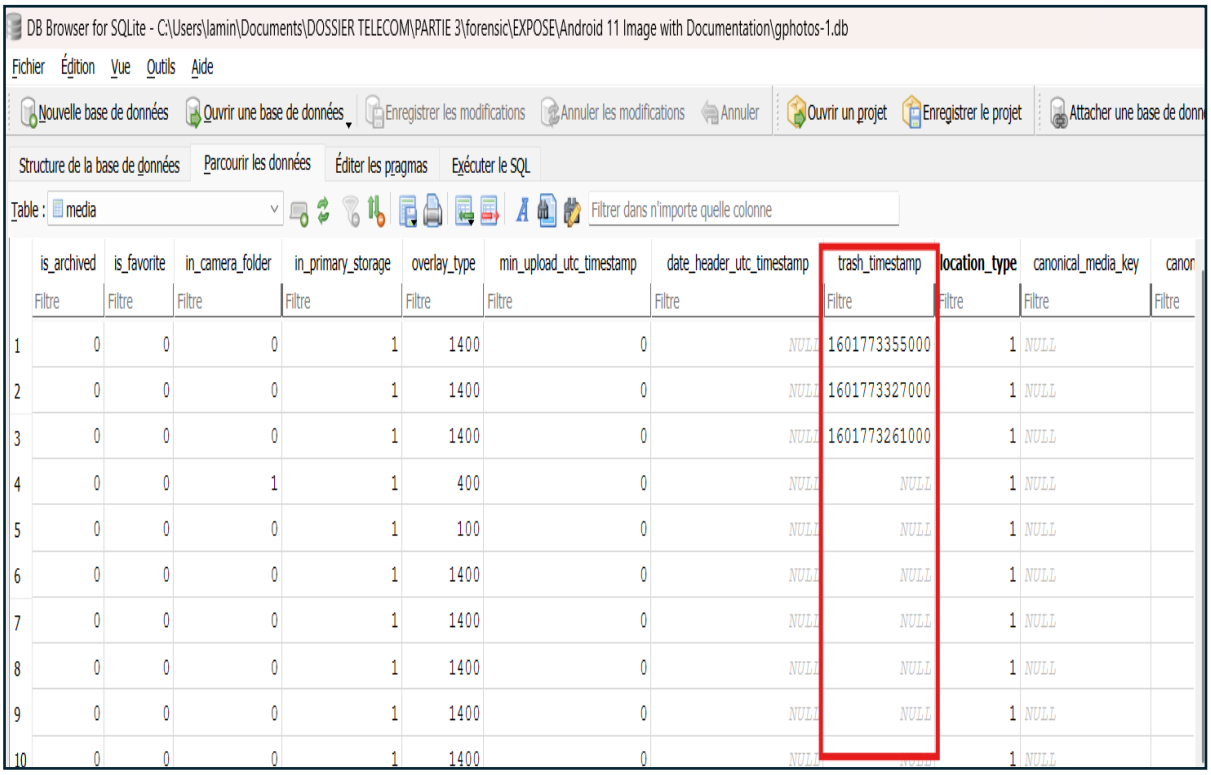
Nouvelle base de données | Ouvrir une base de données | Enregistrer les modifications | Annuler les modifications | Annuler | Ouvrir un projet | Enregistrer le projet

Structure de la base de données | Parcourir les données | Éditer les pragmas | Exécuter le SQL

Table : media — Filtrer dans n'importe quelle colonne

	_id	dedup_key	utc_timestamp	timezone_offset	capture_timestamp	month_random_timestamp	type	is_deleted	is_vr	has_local	is_
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filt
1	293	PuM-cEa-aTV6z_-aS1MY-Joh5eU	1601773210308	-14400000	1601758810308	0	1	1	0	0	
2	292	bCul22n7s7aJOIw7oQcuCs0gaz8	1601773107007	-14400000	1601758707007	0	1	1	0	0	
3	291	ZZn7YFRCNFwdfh5EuqmngNvjYgk	1601772592847	-14400000	1601758192847	0	1	1	0	0	
4	410	1etFefgZ0qgHnX6W1Au0kw6tPAI	1601823311000	-14400000	1601808911000	0	2	0	0	1	
5	294	rb23lopR9i2aLZYl84jU_-5RQN4	1601778746000	-14400000	1601764346000	0	1	0	0	1	
6	298	vAiGxC3WDKcwS39ebX6zHQtiIHE	1601773227000	-14400000	1601758827000	0	1	0	0	1	
7	299	_q_0sNM_wGquyGvC8hF-w89lUg0	1601773136000	-14400000	1601758736000	0	1	0	0	1	
8	300	jAxj35EhbFpIj_CJYTnmPIYNVAo	1601773126000	-14400000	1601758726000	0	1	0	0	1	
9	301	3KiZ8N2HNpAZrd5OlCNoWc2qpPk	1601773068000	-14400000	1601758668000	0	1	0	0	1	
10	302	VqWq0uLWlsPPhdAxXyufmvdYGw8	1601773038000	-14400000	1601758638000	0	1	0	0	1	
11	303	evt8lU1Oy8qfCdUls3UdbD_Jgik	1601773026000	-14400000	1601758626000	0	1	0	0	1	
12	304	zlLJ7Drx65SE6HF87Tvj0Y_RfVk	1601770161000	-14400000	1601755761000	0	1	0	0	1	
13	305	3xJ6a9p4h1uxz4ACKGfn6jvR8_Y	1601769873000	-14400000	1601755473000	0	1	0	0	1	
14	306	Mbj4ZH8gC-IJ27QftDOcibmy88M	1601766862000	-14400000	1601752462000	0	1	0	0	1	
15	283	yZy7lBn6947riXdmABrmTaLvjHA	1601735064000	-14400000	1601720664000	0	1	0	0	1	

Figure 26 : Les champs "dedup_key" et "is_delected" dans la dable media, preuve de suppression



DB Browser for SQLite - C:\Users\lamin\Documents\DOSSIER TELECOM\PARTIE 3\forensic\EXPOSE\Android 11 Image with Documentation\gphotos-1.db

Fichier Édition Vue Outils Aide

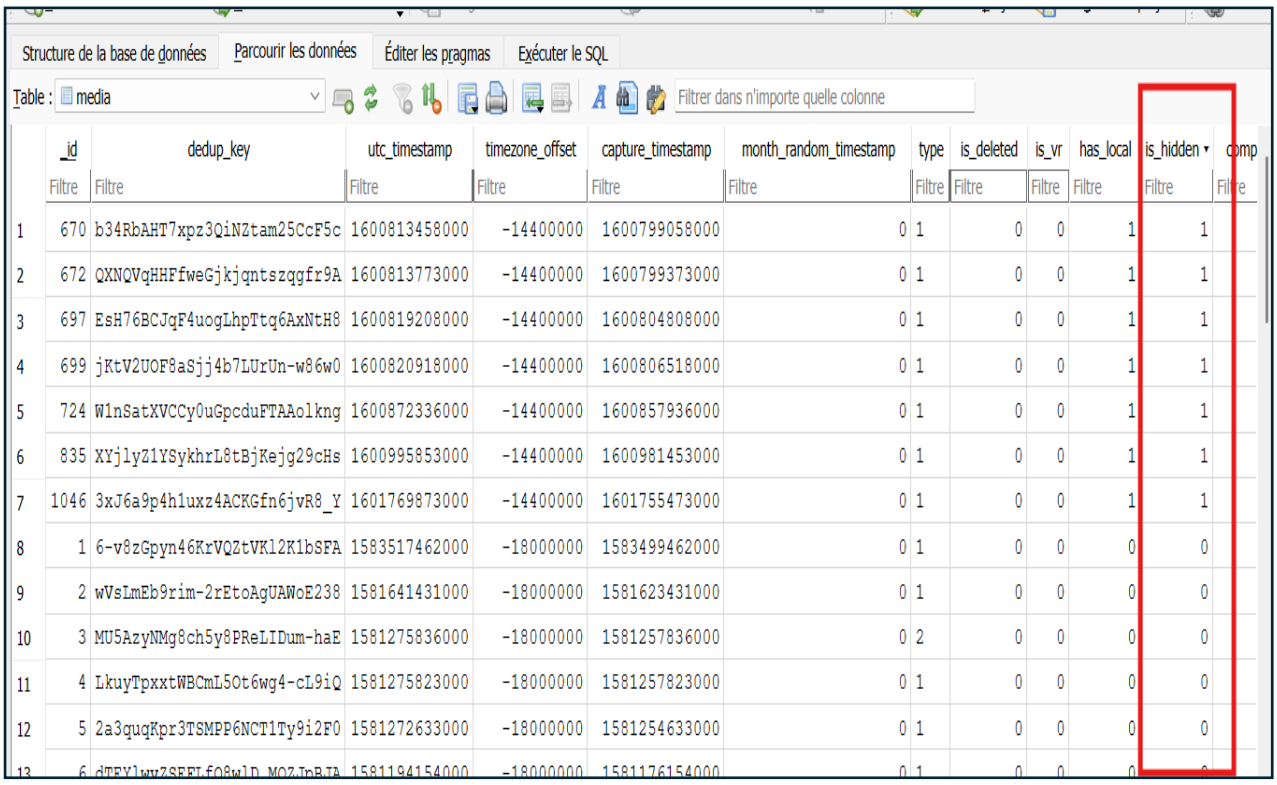
Nouvelle base de données | Ouvrir une base de données | Enregistrer les modifications | Annuler les modifications | Annuler | Ouvrir un projet | Enregistrer le projet | Attacher une base de donn

Structure de la base de données | Parcourir les données | Éditer les pragmas | Exécuter le SQL

Table : media — Filtrer dans n'importe quelle colonne

	is_archived	is_favorite	in_camera_folder	in_primary_storage	overlay_type	min_upload_utc_timestamp	date_header_utc_timestamp	trash_timestamp	location_type	canonical_media_key	canor
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	0	0	0	1	1400	0	NULL	1601773355000	1	NULL	
2	0	0	0	1	1400	0	NULL	1601773327000	1	NULL	
3	0	0	0	1	1400	0	NULL	1601773261000	1	NULL	
4	0	0	1	1	400	0	NULL	NULL	1	NULL	
5	0	0	0	1	100	0	NULL	NULL	1	NULL	
6	0	0	0	1	1400	0	NULL	NULL	1	NULL	
7	0	0	0	1	1400	0	NULL	NULL	1	NULL	
8	0	0	0	1	1400	0	NULL	NULL	1	NULL	
9	0	0	0	1	1400	0	NULL	NULL	1	NULL	
10	0	0	0	1	1400	0	NULL	NULL	1	NULL	

Figure 26: trush_timestamp , temps de suppression des fichiers



	_id	dedup_key	utc_timestamp	timezone_offset	capture_timestamp	month_random_timestamp	type	is_deleted	is_vr	has_local	is_hidden	comp
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	670	b34RbAHT7xpz3QiNZtam25CcF5c	1600813458000	-14400000	1600799058000	0	1	0	0	1	1	
2	672	QXNQVqHHFfweGjkjqntszqgfr9A	1600813773000	-14400000	1600799373000	0	1	0	0	1	1	
3	697	EsH76BCJqF4uogLhpTtq6AxNtH8	1600819208000	-14400000	1600804808000	0	1	0	0	1	1	
4	699	jKtV2UOF8aSjj4b7LUrUn-w86w0	1600820918000	-14400000	1600806518000	0	1	0	0	1	1	
5	724	W1nSatXVCCy0uGpcduFTAAolkng	1600872336000	-14400000	1600857936000	0	1	0	0	1	1	
6	835	XYjlyZ1YSykhrL8tBjKejg29cHs	1600995853000	-14400000	1600981453000	0	1	0	0	1	1	
7	1046	3xJ6a9p4h1uxz4ACKGfn6jvR8_Y	1601769873000	-14400000	1601755473000	0	1	0	0	1	1	
8	1	6-v8zGpyn46KrVQZtVKl2K1bSFA	1583517462000	-18000000	1583499462000	0	1	0	0	0	0	
9	2	wVsLmEb9rim-2rEtoAgUAWoE238	1581641431000	-18000000	1581623431000	0	1	0	0	0	0	
10	3	MU5AzyNMg8ch5y8PReLIDum-haE	1581275836000	-18000000	1581257836000	0	2	0	0	0	0	
11	4	LkuyTpxxtWBCmL5Ot6wg4-cL9iQ	1581275823000	-18000000	1581257823000	0	1	0	0	0	0	
12	5	2a3quqKpr3TSMPP6NCT1Ty9i2F0	1581272633000	-18000000	1581254633000	0	1	0	0	0	0	
13	6	dTFYlwyZSFFLfO8wlD_MOZJpBJA	1581194154000	-18000000	1581176154000	0	1	0	0	0	0	

Figure 28 27: La colonne is hidden contient des valeurs non nulles

Tables des matières

Bibliographie

https://app.letsdefend.io/training/lesson_detail/evidentiary-data-on-android/

https://digitalcorpora.s3.amazonaws.com/corpora/mobile/android_11.zip

https://www.reddit.com/r/archlinux/comments/pgoa4t/a_word_of_warning_about_f2fs/?tl=fr

https://www.patricelaurent.net/android-booster-performances-f2fs/

https://www.ninjaone.com/fr/it-hub/endpoint-security/chiffrement-complet-du-disque/#:~:text=Prot%C3%A8ge%20les%20donn%C3%A9es%20sensibles&text=Le%20FDE%20contribue%20%C3%A0%20r%C3%A9duire,stock%C3%A9es%20les%20donn%C3%A9es%20de%20sauvegarde%20

https://www.tinymdm.fr/chiffrement-android-entreprise/

https://fastercapital.com/fr/mots-cle/based-encryption-fbe.html