black hat MIDDLE EAST AND AFRICA

26 - 28 NOVEMBER 2024
RIYADH, SAUDI ARABIA

DriveFS Sleuth

Your Ultimate Google Drive File Stream Investigator!

Presented BY: Amged Wageh

tahaluf an informa company

الاتحاد السعودي للأمن السيبراني والبرمجة والدرونز
SAUDI FEDERATION FOR CYBERSECURITY, PROGRAMMING & DRONES

black hat

black hat
MIDDLE EAST AND AFRICA

Amged Wageh

Incident Response Specialist - Kaspersky

Ex: Sr. DFIR Consultant - Cisco

DriveFS Sleuth Author

linkedin.com/in/amgedwageh/

amged_wageh@outlook.com

black hat
MIDDLE EAST AND AFRICA

Agenda

Google Drive Applications

Motivation and Research Objective

Research Findings

DriveFS Sleuth – The automation tool

Google Drive for Desktop
A file syncing application that helps storing and accessing files from Google **Drive** and Google **Photos**.

File Stream!
Is a feature to **stream files on demand**, avoiding the need to occupy disk space by downloading all files preemptively. Users can access all previously synced files through Windows Explorer, without the necessity of storing an offline version.

Motivation and Research Objectives

Frequent data leakage
A lot of classified documents were frequently leaked to Scribd!

Hard to Investigate
Insider threats are usually more difficult to detect, especially if the leaking person has justified access to the leaked documents.

Google Drive App. was in common
The suspected users had google drive for desktop application installed.

Challenge!
No clue how to investigate them.

What is expected from a successful investigation?

Syncing Accounts
Which accounts were used?
What if an account logged off?

Synced files context
What are the synced files?
What are their attributes?

Deleted Files
Can we know about them?

Recovery
Can we recover interesting files from the cache?

black hat
MIDDLE EAST AND AFRICA

Research Findings

What happens upon installation?

During its installation, Google Drive for Desktop configures the system for operational functionalities, **these system modifications can serve as indicators to detect unauthorized installation of the application.**

Registry

- Installation information
- CLSIDs for COM objects
- Shell Icon Overlays
- Shell Extensions
- Outlook Add-ons
- Persistence via Run

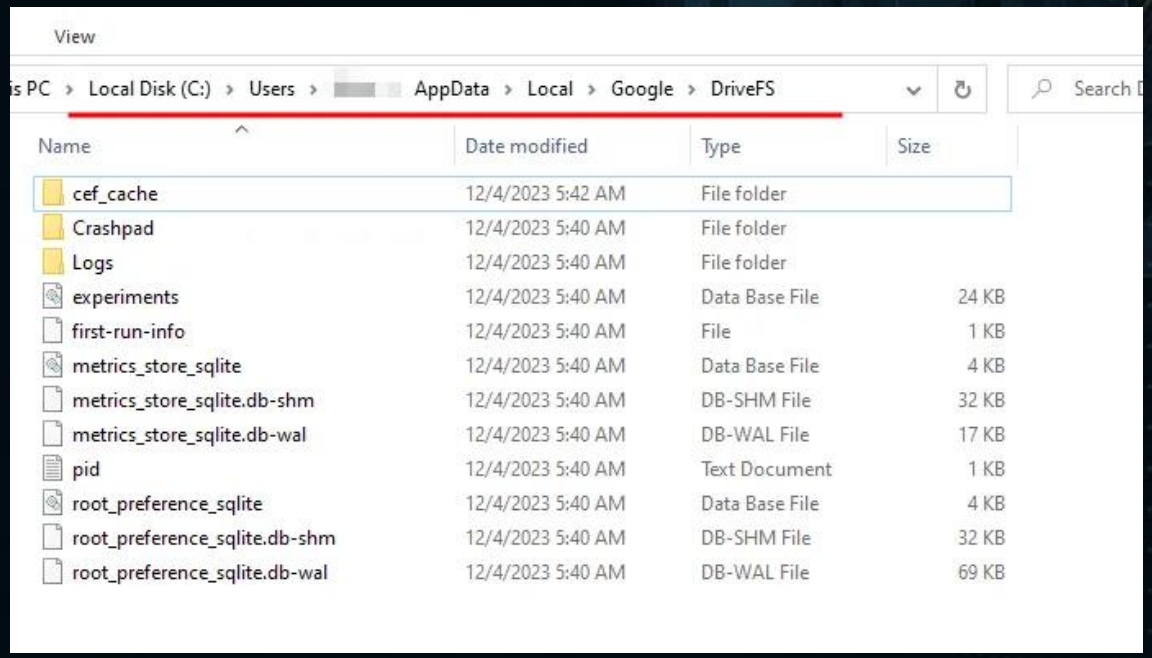
File system

- Installation files, icons, DLLs, etc.
- Start Menu Ink
- ...

https://amgedwageh.medium.com/drivefs-sleuth-investigating-google-drive-file-streams-disk-artifacts-0b5ea637c980#:~:text=What%20happens%20upon%20installation%3F

What happens upon the application's first launch?

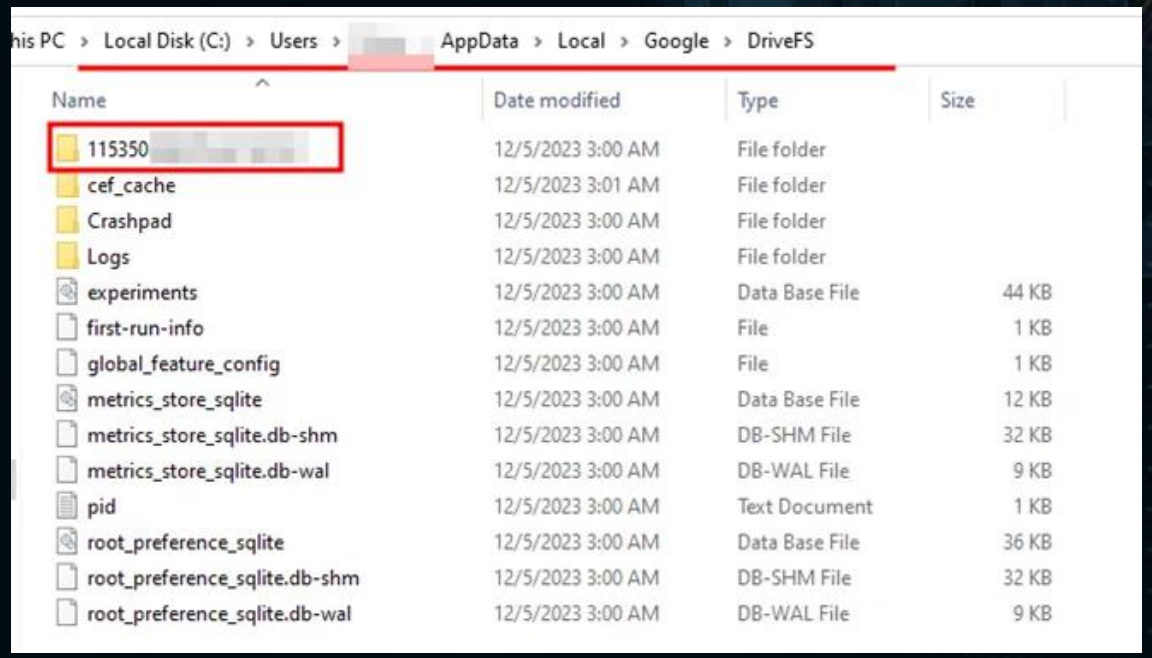
Many files for tracking the features' configuration are written by default at `%LocalAppData%\Google\DriveFS`



What happens upon logging in?

For each user login, a directory is created named with the user's account ID. This folder provides valuable insights into the items that the user has synchronized with his account and the attributes of the items.

***If the profile directory doesn't exist, then the account is not currently logged in.*



What is the nature of the artifacts?

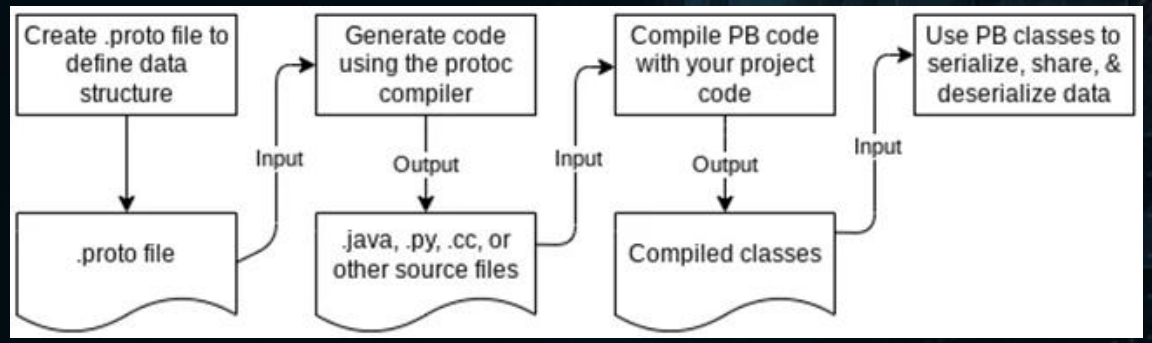
Log files
- txt files.
- Not very crucial, contains important information though.

SQLite Databases
- Not encrypted.
- Stores a lot of enum values (needs experiments to figure them out).
- Depends on protobuf for storing important hidden info. (.proto files are not available)

What happens upon the application's first launch?

Protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data.

`.proto` files are definition language files, the code that the proto-compiler generates to interface with data.
protobuf messages do not inherently carry information about their structure.



ref: https://protobuf.dev/overview/

Heads-Up!

Keep in mind that these artifacts are mostly for functional purposes of the application itself on the host machine so most of these artifacts are only stored offline and not being synced. Consequently, **when conducting investigations on the same account through two different triages from two different machines, variations in results may arise. The variations will not affect the investigation integrity.**

black hat
MIDDLE EAST AND AFRICA

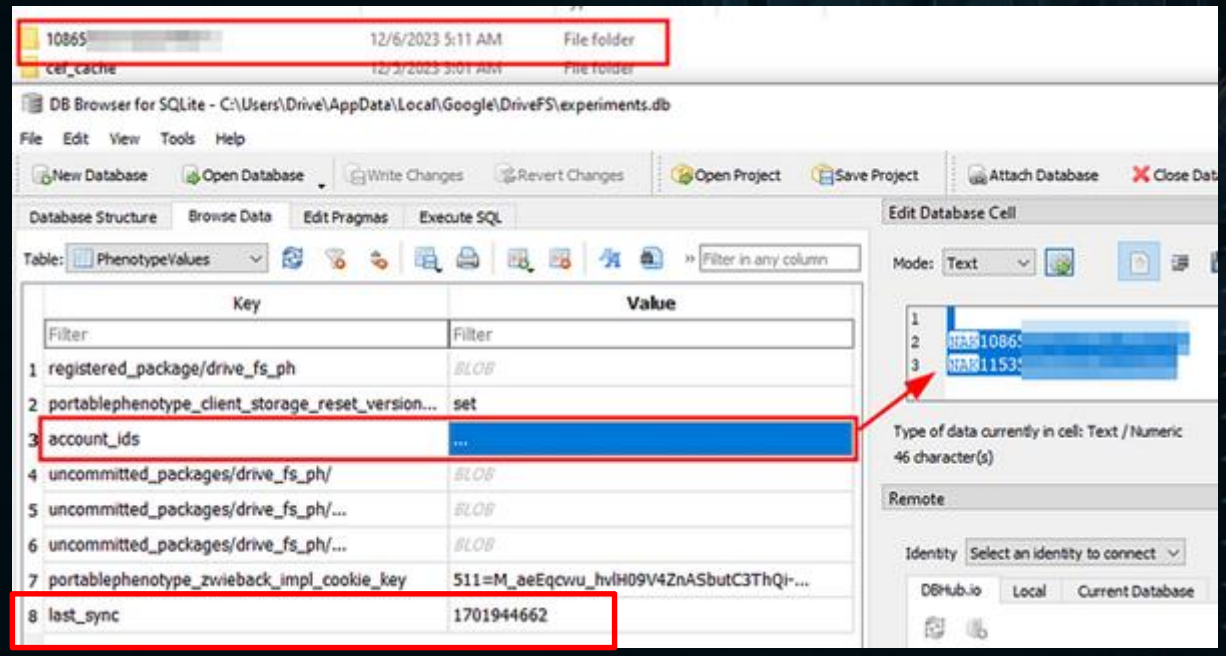
Research Findings

Installation Context

Which accounts were used?

The `experiments.db` stores the IDs of the logged-in accounts in the `account_ids` value in the `PhenotypeValues` table. The account IDs are not removed after logging out.
The `experiments.db` exists in `%LocalAppData%\Google\DriveFS\experiments.db`

The same table contains an epoch timestamp that represents last syncing time in the `last_sync` value.

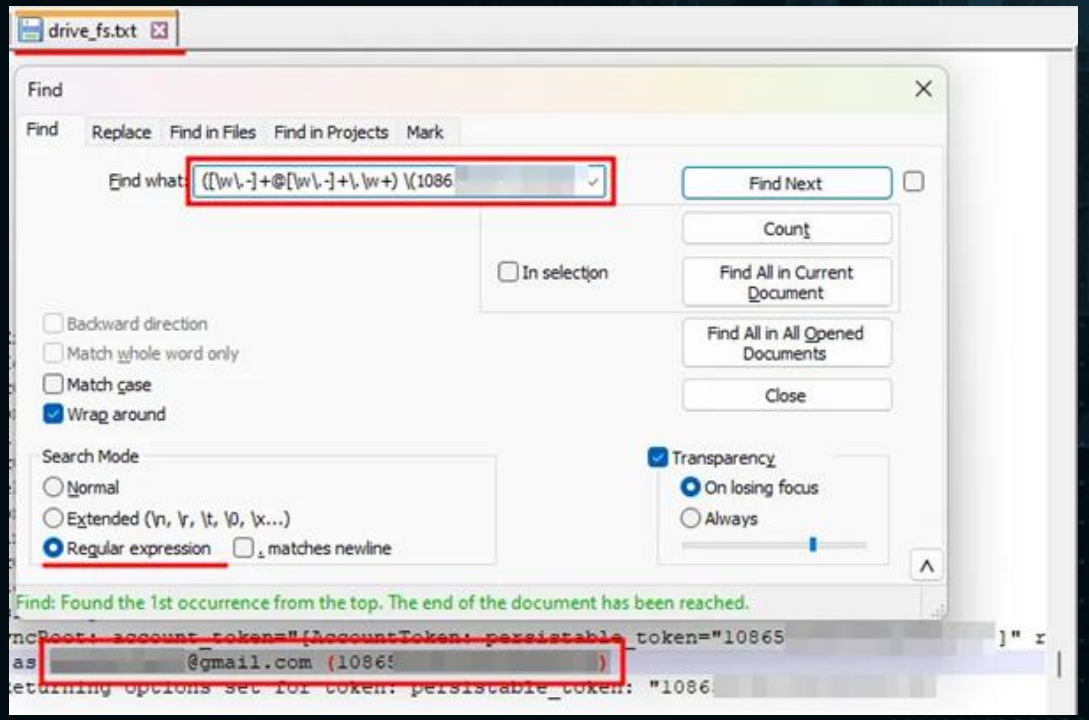


Who owns this ID?

The drive_fs[_num].txt log files mentions the email and the corresponding ID in the following format **abc@gmail.com (1651...)**.

The drive_fs[_num].txt log files exists in `%LocalAppData%\Google\DriveFS\Logs\drive_fs[_num].txt`

We can search by the this regex `([\w\.-]+@[\w\.-]+\.\w+) \(<account_id>\)`



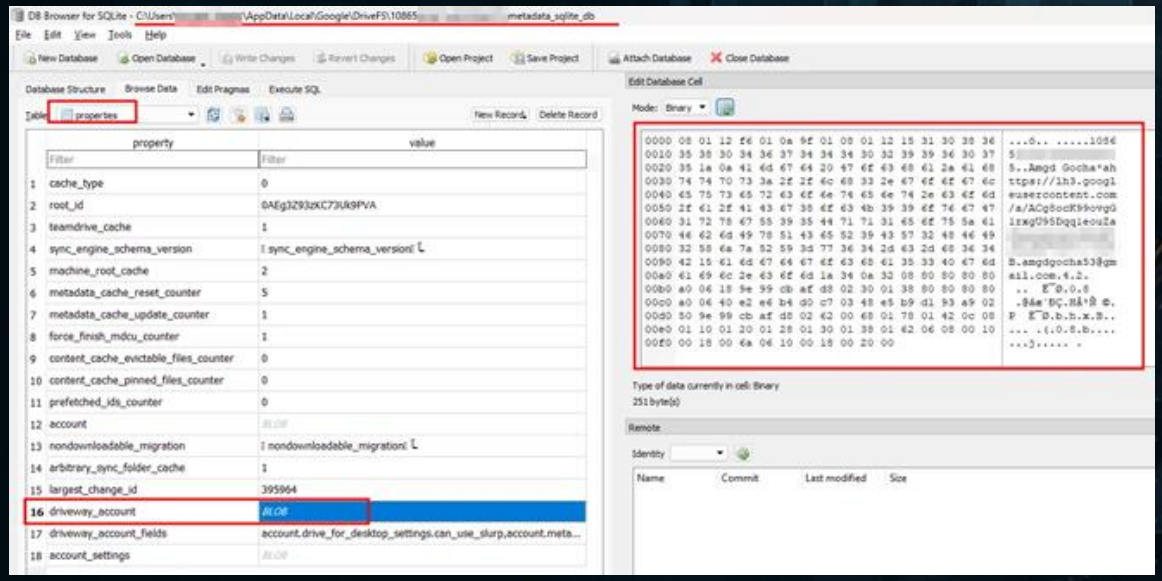
Some enrichment...

Under the profile directory, the `metadata_sqlite_db` database contains a properties table which contains `driveway_account` and account properties.

Both contains the same information *(backward compatibility)*.

Data are stored as a protobuf.

We can determine the **display name, account photo, account email,** and many more.

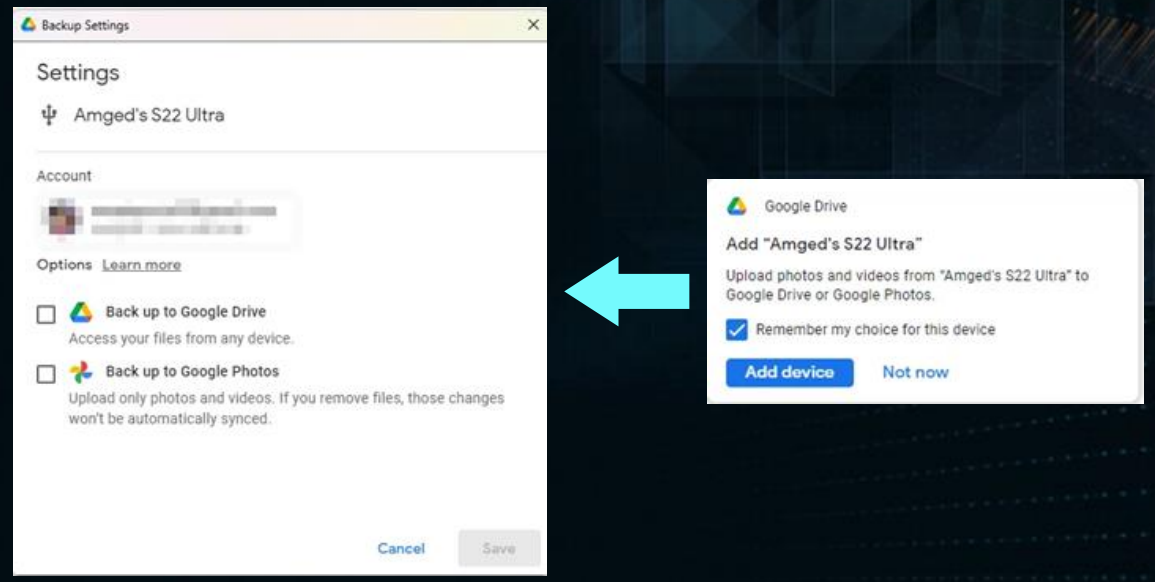



```
root:
    1 <varint> = 1
    2 <chunk> = message:
        1 <chunk> = message:
            1 <varint> = 1
            2 <chunk> = "10865
            3 <chunk> = "Amgd Gocha"
            5 <chunk> = "https://lh3.googleusercontent.com/a/ACg8ocK99
            8 <chunk> = "amgdgocha53@gmail.com"
        3 <chunk> = message:
```

Media storages syncing

A popup window is shown to check enabling the syncing from a storage device when it's connected.

A follow-up window will be shown to configure the syncing service (drive or photos) and the syncing account.

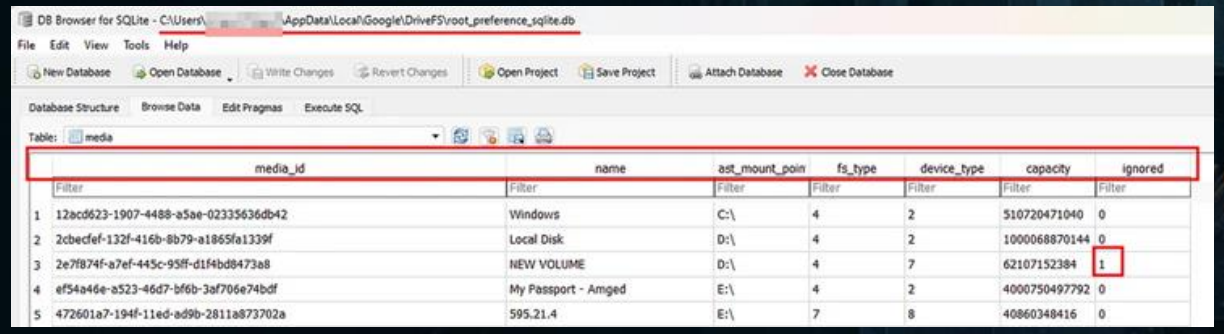


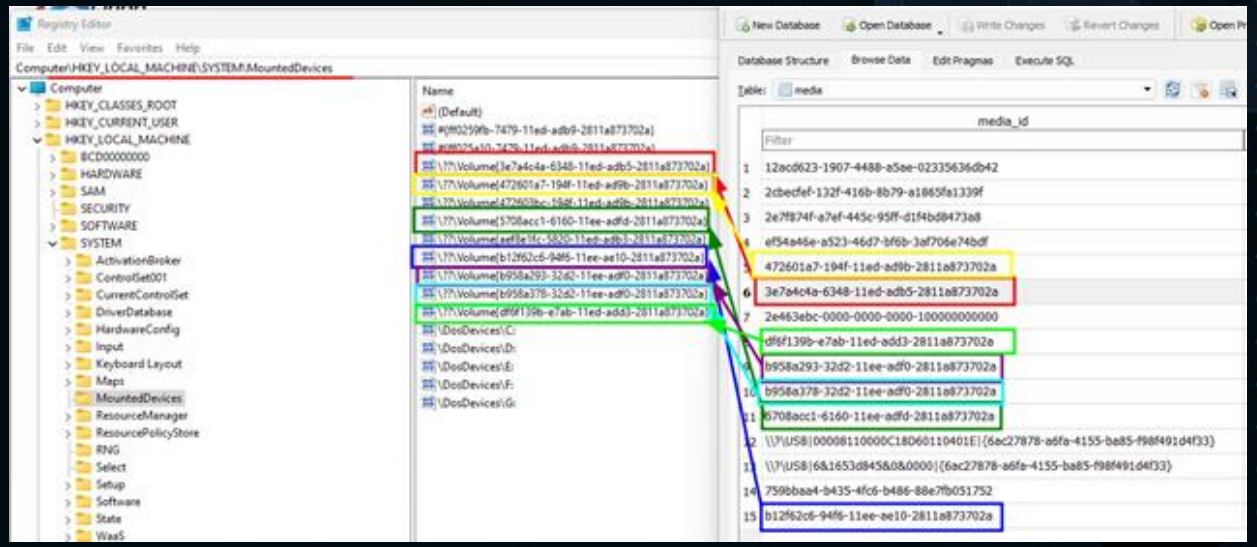
Media storages syncing

Accordingly, a record will be stored in the `root_preference_sqlite.db` in the media table.

The `capacity` will be set to `-1` if the connected device didn't allow access.

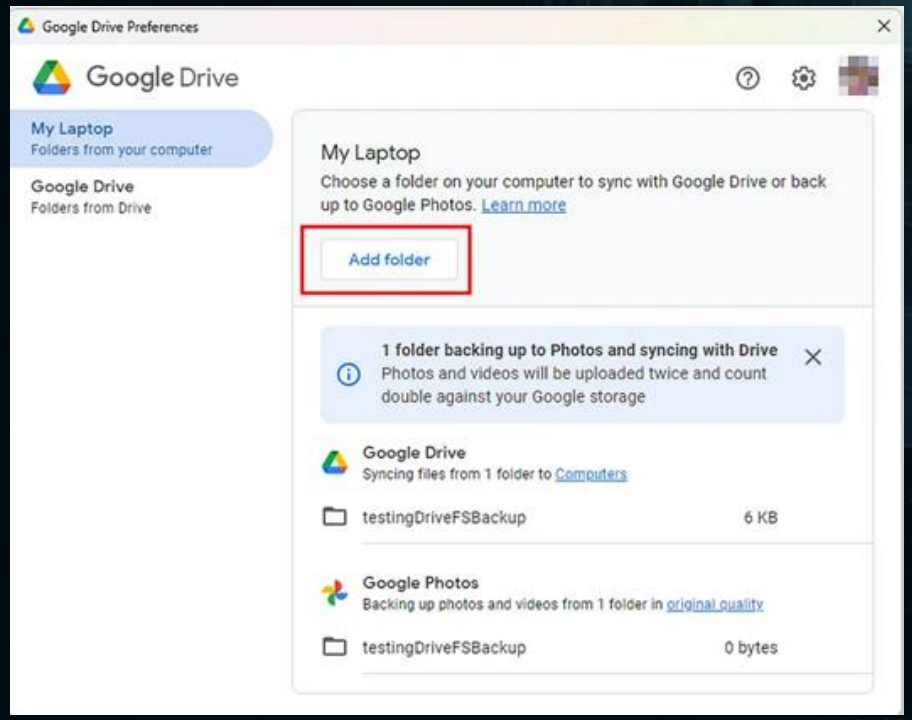
The `media_id` contains the same GUID stored in the registry.





Mirroring roots

Mirroring roots can be configured so that any modification to a specific synced directory will be reflected in both the local and cloud versions.

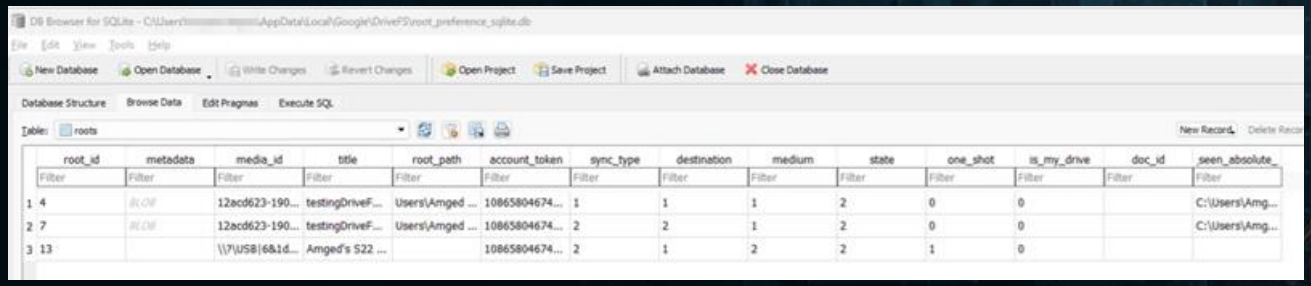


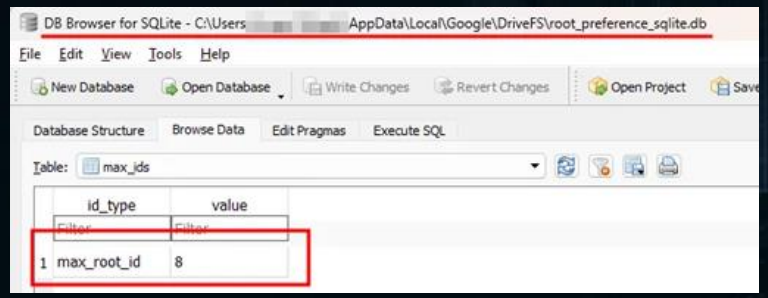
Mirroring roots

Stored in the roots table in the `root_preference_sqlite.db` where an incrementing id is assigned to each root.

Destination equals 1 to sync with Drive, and it equals 2 to sync with Photos. Two entries will be written to sync with both.

The max root id is stored in the `max_ids` table in the same DB. If the `max_root_id` > the number of roots in the roots table, then some roots have been removed or modified.





The `one_shot` column stores 1 if the user didn't select the "Remember my choice for this device." option and the record will be deleted once unplugged, and 0 otherwise.

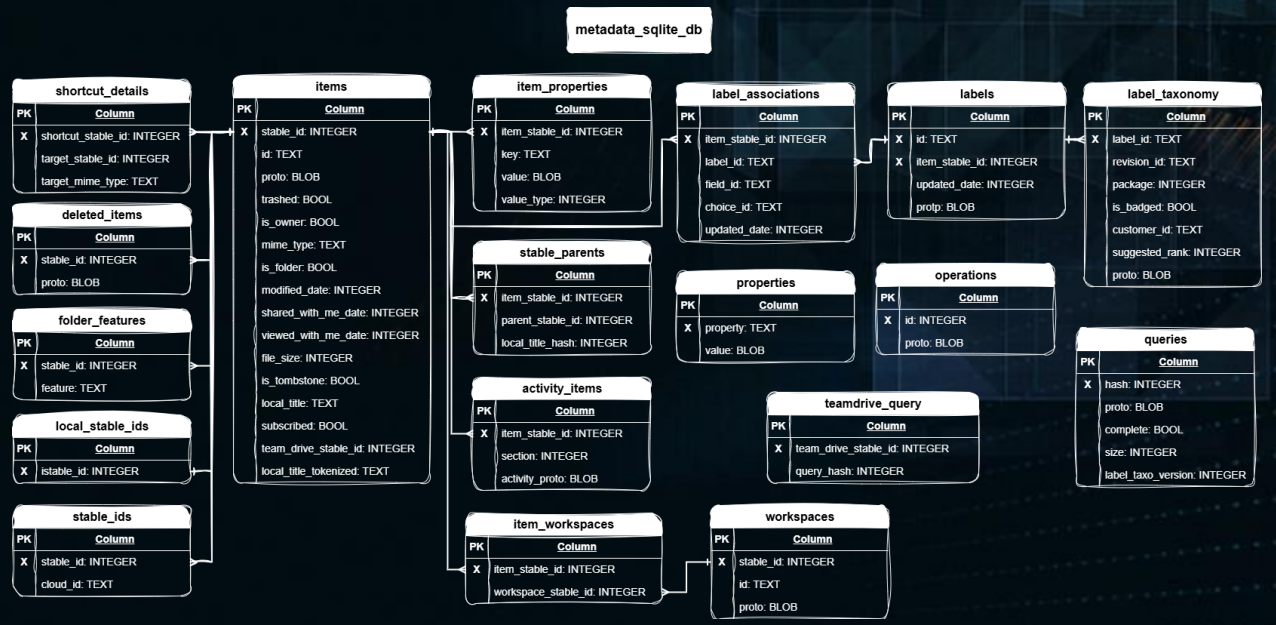
black hat
MIDDLE EAST AND AFRICA

Research Findings

Accounts Context

What are the synced items?

The `metadata_sqlite_db` stands out as the pivotal database, containing crucial data regarding synchronized, deleted, and shared items with the user, it consists of 18 tables. A copy of this DB exists in the `mirror_metadata_sqlite.db`

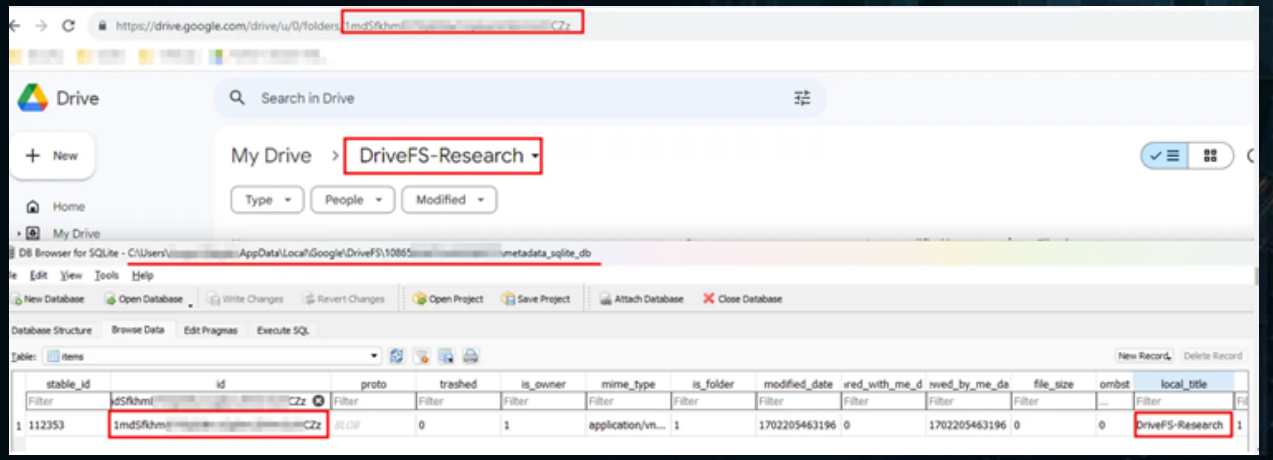


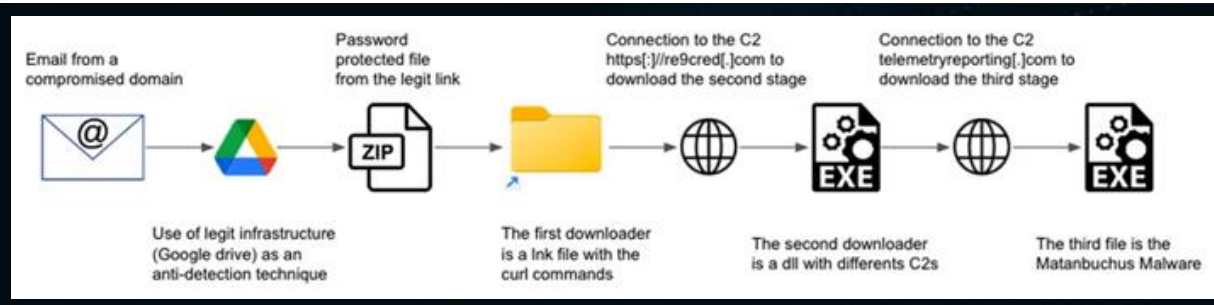
It exists under the user's profile `%LocalAppData%\Google\DriveFS\<account_id>\metadata_sqlite_db`

What are the different IDs?

An internal ID called `stable_id (item_id)` is assigned to each item and it works as the primary and foreign keys for the tables relationships.

The `id` column tracks the URL ID of the item. Very important for scoping.





Matanbuchus malware attack flow
Ref: https://intelligence.abnormalsecurity.com/blog/google-drive-matanbuchus-malware

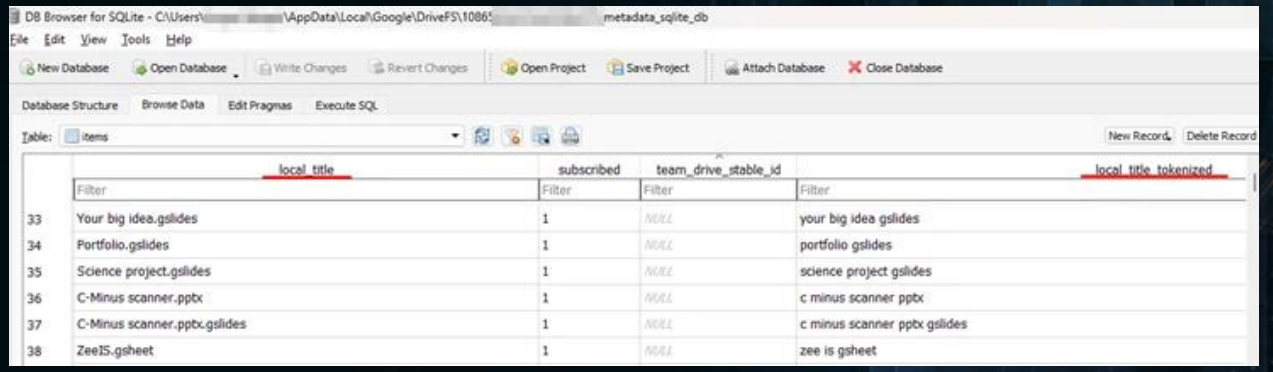
What are the items info.?

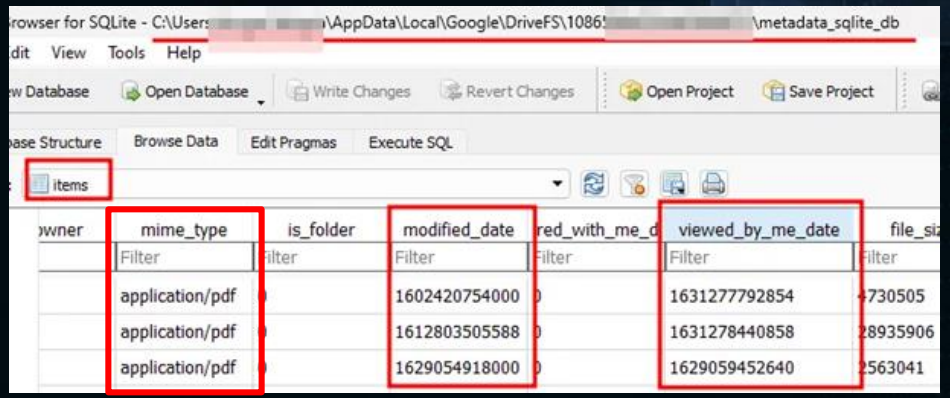
`local_title` and `local_title_tokenized` for the item name.

`modified_date` for the last modification time

`viewed_by_me_date` for the last time when the user viewed the item.

`mime_type` of the item.

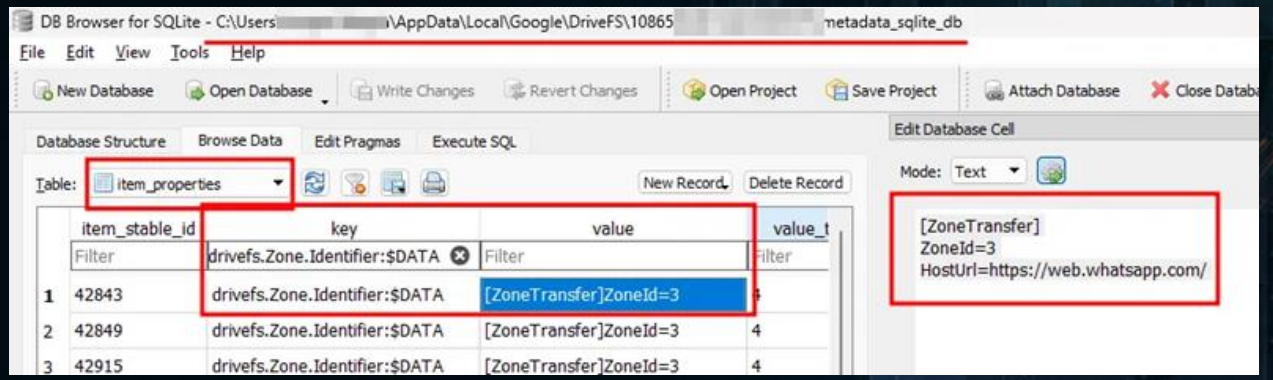




What about file system metadata?

The `item_properties` table stores additional item's metadata and it may include file system metadata as well.

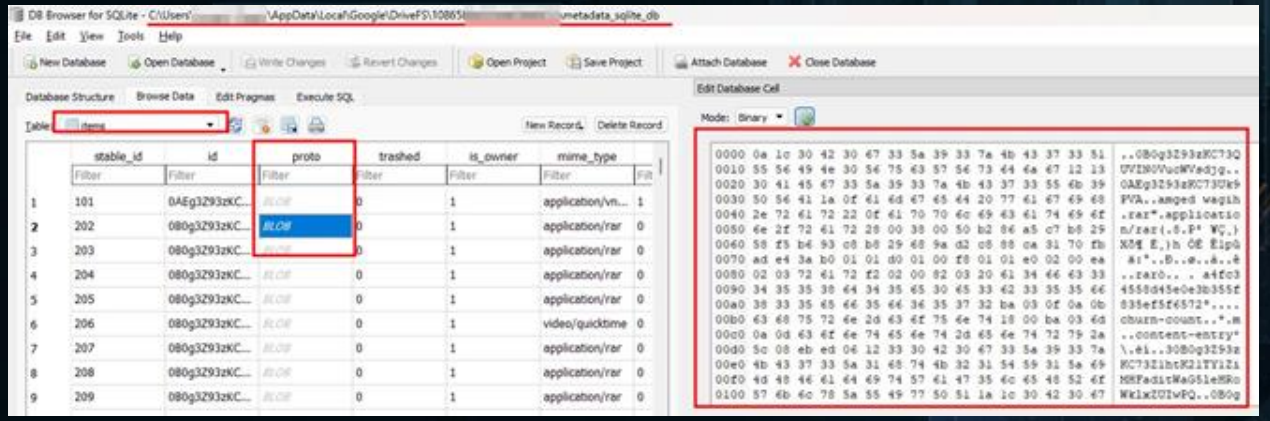
Sometime the ADS can be preserved as well!



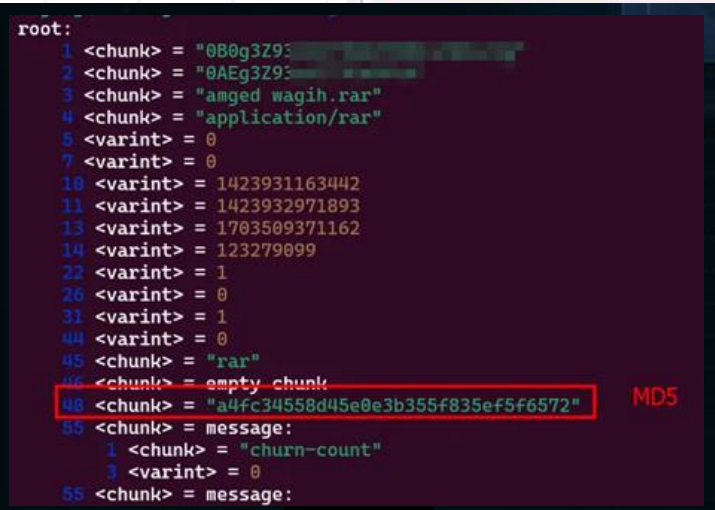
For more info: https://amgedwageh.medium.com/drivefs-sleuth-investigating-google-drive-file-streams-disk-artifacts-0b5ea637c980#:~:text=The%20list%20of,across%20additional%20properties.

Revealing hidden info. – MD5 hash?

The proto attribute in the items table contains a blob of a protobuf message.



The protobuf contains the same information that exists in the database plus other hidden info, the most important one is the MD5 hash of the item!



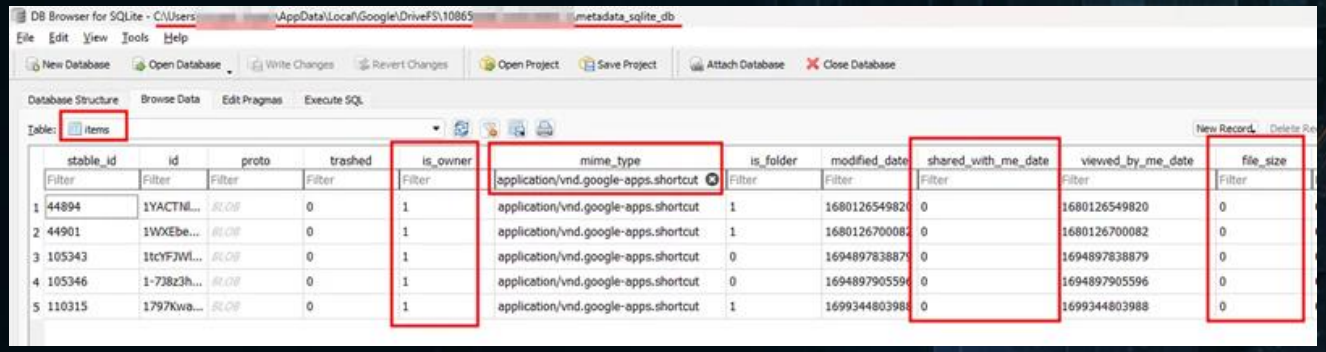
We are able to determine the hash even without the file being cached!

What are the shortcut files?

Drive allows users to create shortcuts for items (shared or owned) to quickly access them.

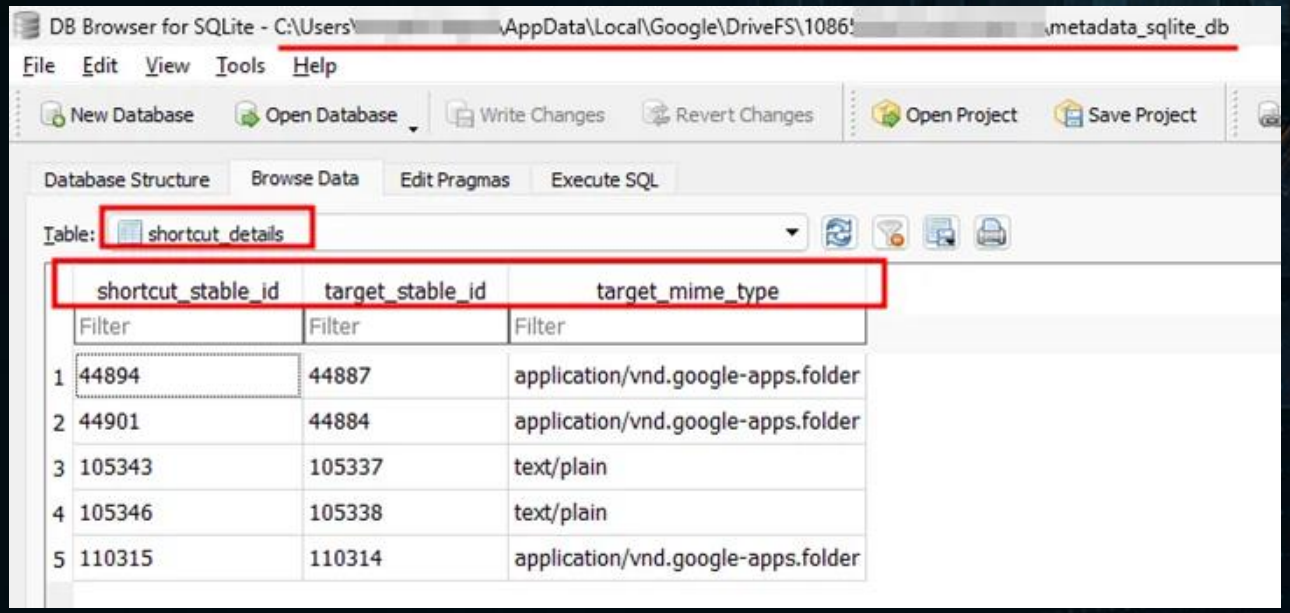
`mitme_type` is `application/vnd.google-apps.shortcut`.

`File_size` will always be `0`.



To which item does the shortcut refer?

The `shortcut_details` table stores the shortcut item id in the `shortcut_stable_id` and the id of item it refers to in the `target_stable_id`, the type of the target file is stored in the `target_mime_type`.

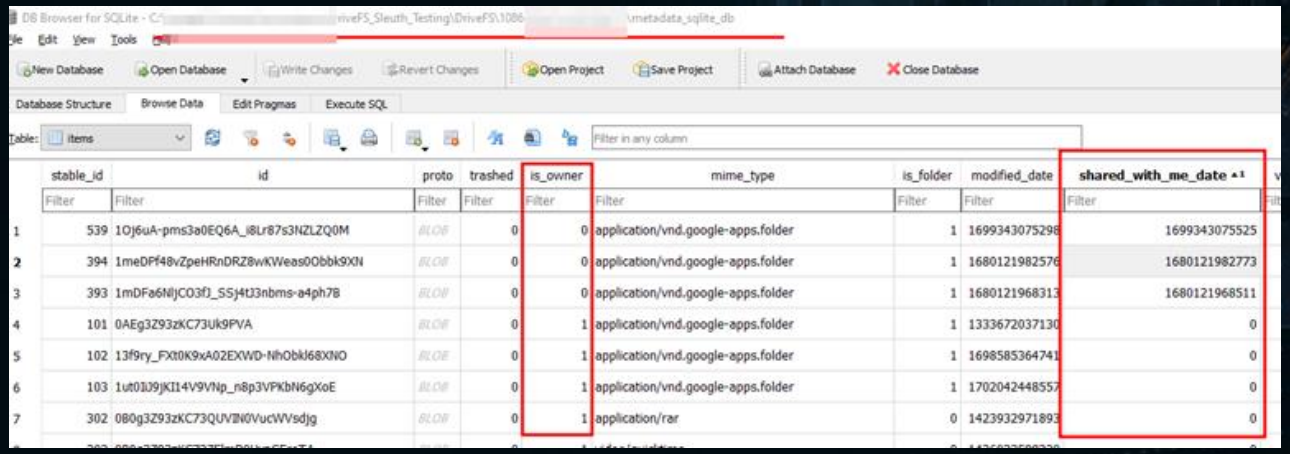


Who owns the item?

The `is_owner` attributes equals `1` for the items that the user owns and `0` otherwise.

The `shared_with_me_date` contains an epoch timestamp of the sharing date.

The user always owns the shortcut item that links to the item shared with him.



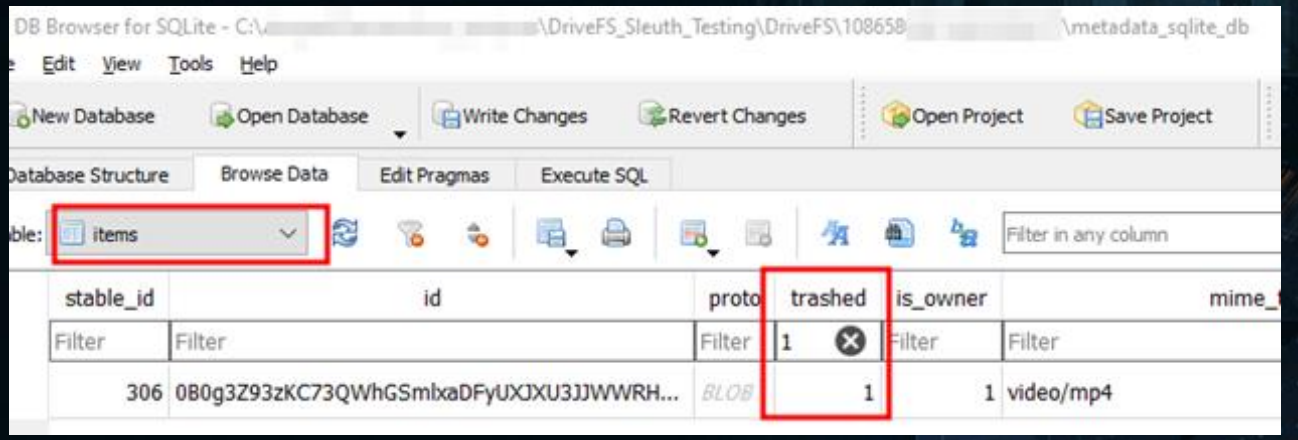
	stable_id	id	proto	trashed	is_owner	mime_type	is_folder	modified_date	shared_with_me_date
1	539	1OJ6uA-pms3a0EQ6A_i8Lr87s3NZLZQ0M	BLOB	0	0	application/vnd.google-apps.folder	1	1699343075298	1699343075525
2	394	1meDPf48vZpeHRnDRZ8wKWeas0Obbk9XN	BLOB	0	0	application/vnd.google-apps.folder	1	1680121982576	1680121982773
3	393	1mDFa6NljCO3fJ_SSJ4tJ3nbms-a4ph7B	BLOB	0	0	application/vnd.google-apps.folder	1	1680121968313	1680121968511
4	101	0AEg3Z93zKC73Uk9PVA	BLOB	0	1	application/vnd.google-apps.folder	1	1333672037130	0
5	102	13f9ry_FXt0K9xA02EXWD-NhObkl68XNO	BLOB	0	1	application/vnd.google-apps.folder	1	1698585364741	0
6	103	1ut01J9jKI14V9VNp_n8p3VPKbN6gXoE	BLOB	0	1	application/vnd.google-apps.folder	1	1702042448557	0
7	302	0B0g3Z93zKC73QUVlN0VucWVsdjg	BLOB	0	1	application/rar	0	1423932971893	0

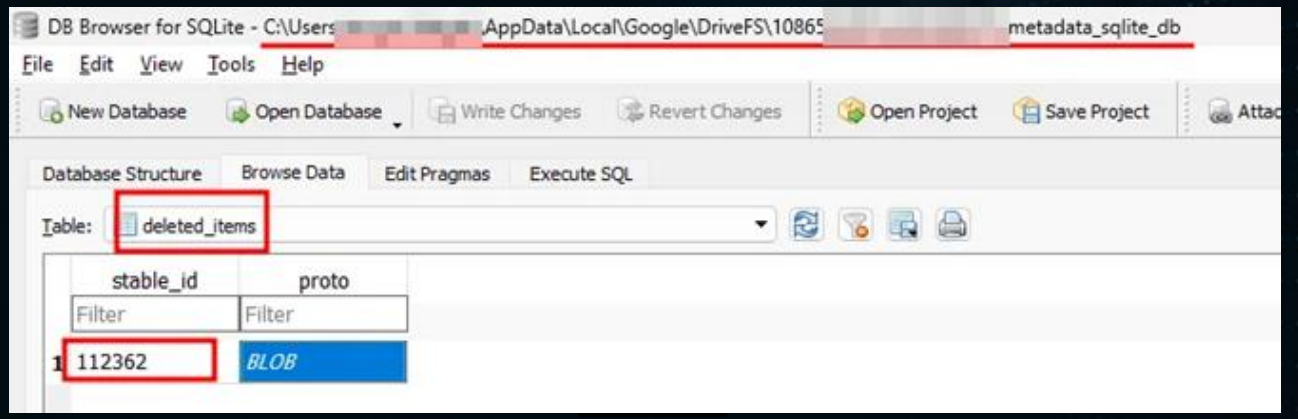
What happens with deleted files?

When an item is deleted, it undergoes a 30-day retention period in the Trash, allowing the potential for restoration.

The trashed attribute in the items table will be set to 1 for trashed items and 0 otherwise.

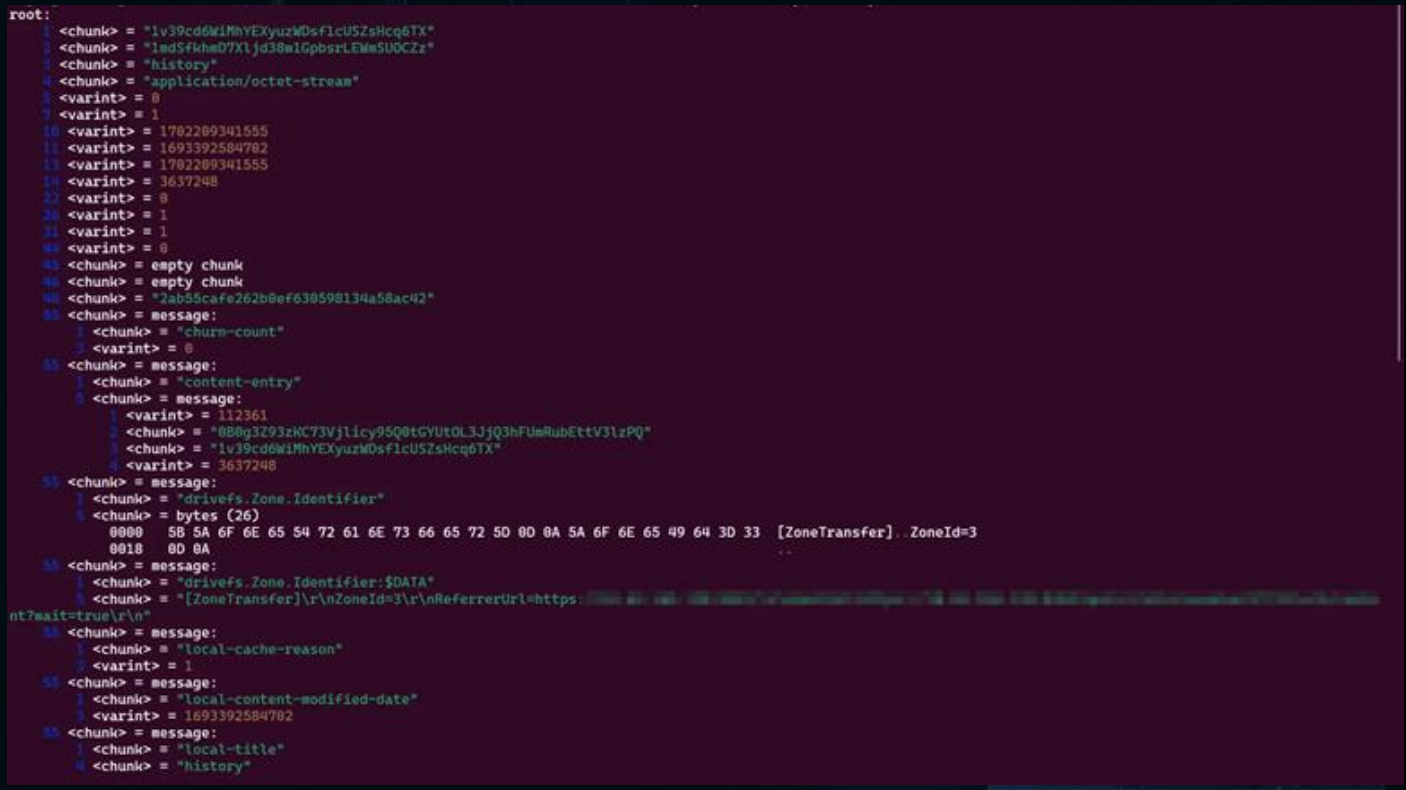
Upon permanent deletion, the record is removed from the items table and a new record is stored in the deleted_items table.





Can we recover them?

Absolutely! by decoding the protobuf message.



Revealing the proto scheme...

Attribute	Index	Attribute	Index
url_id	1	modified_date	11
parent_url_id	2	viewed_by_me_date	13
local_title	3	file_size	14
mime_type	4	file extension	45
trashed	7	MD5	48

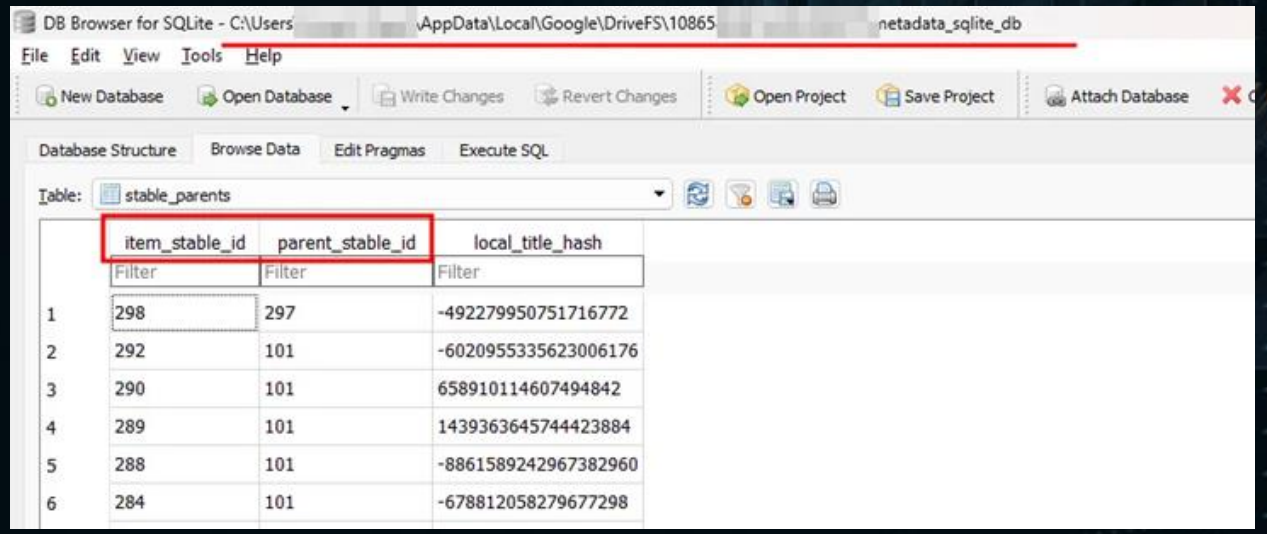
Revealing the proto scheme...

Attribute	Index	Attribute	Index
folder feature	50	is_owner	63
item properties	55	stable_id	88
trashed-locally	55-2	parent_stable_id	89
trashed-locally-metadata	55-5	trashed-locally-name	55-4

What is the hierarchal structure?

The items' parent-child relationship is tracked in the `stable_parent` table where the `item_stable_id` refers to the id of the item and the `parent_stable_id` refers to its parent.

A deleted parent item can be identified when the `parent_stable_id` doesn't refer to any item in the items table. *(happens with shared items and shortcuts).*



black hat
MIDDLE EAST AND AFRICA

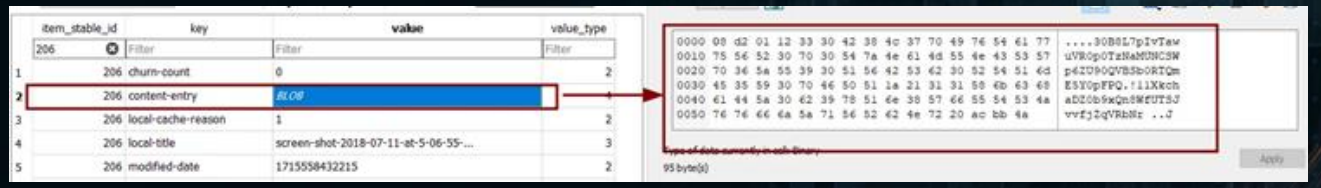
Research Findings

Data Recovery

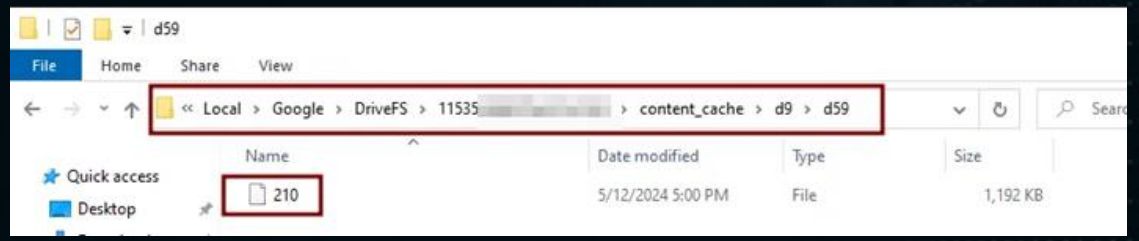
How to recover cached content?

By decoding the protobuf message that is stored in the content-entry property of the item which is stored in the `item_properties` table in the `metada_sqlite_db`, we can get an id.

That id is the filename under the `content_cache` directory.



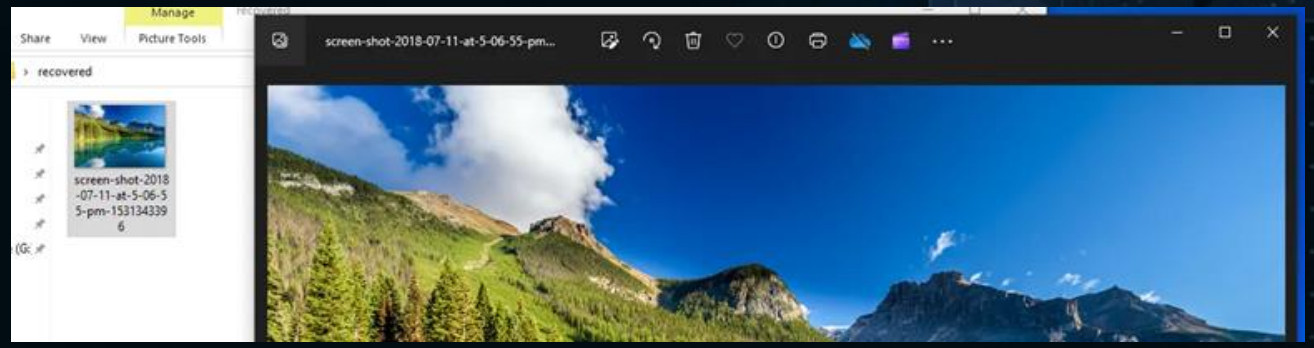

```
PS C:\Users\DriveFS Sleuth> protodeep -t protobuf 'C:\users\DriveFS Sleuth\Desktop\bins\206-content-entry.bin'
Finding the protobuf starting chunk...
[+] Bruteforce index : 0

[1]  1 -> <int> = 210 [1]
[1]  2 -> <string> = "0B8L7pIvTawuVR0p0TzNaMUNCSWp6ZU90QVBSb0RTQmE5Y0pFPQ" [2]
[1]  3 -> <string> = "11XkchaDZ0b9xQn8WfUTSJvvfjZqVRbNr" [3]
[1]  4 -> <int> = 1220012 [4]
```



Recovered item renaming...

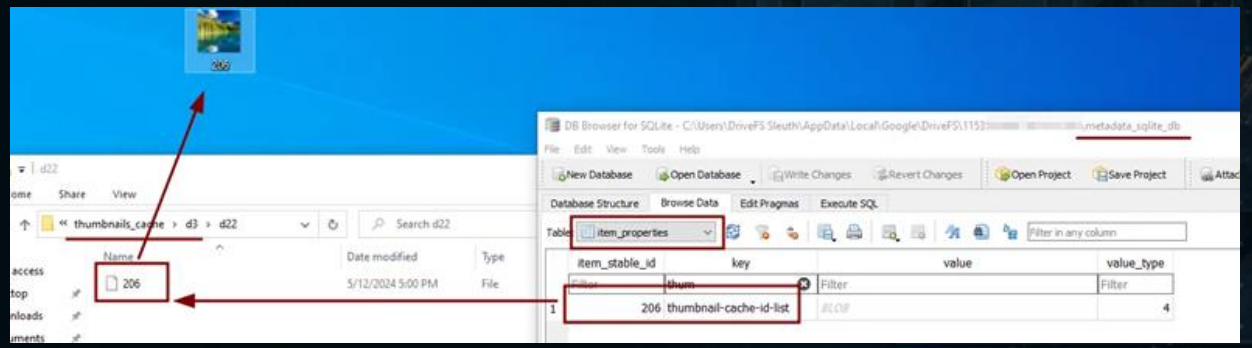
The item attributes can be determined by decoding the item's protobuf.

```
PS C:\Users\DriveFS Sleuth> protodeep -t protobuf 'C:\users\DriveFS Sleuth\Desktop\bins\206-proto.bin'
Finding the protobuf starting chunk...
[+] Bruteforce index : 0

[1]  1 -> <string> = "11XkchaDZ0b9xQn8WfUTSJvvfjZqVRbNr" [1]
[1]  2 -> <string> = "1-By4MCidFv_cFEb-Udv9X7o8Br5xVG6F" [2]
[1]  3 -> <string> = "screen-shot-2018-07-11-at-5-06-55-pm-1531343396.png" [3]
[2]  4,13 -> <fixed64> = 7453017782211797357 [4,13]
[1]  5 -> <int> = 0 [5]
[1]  7 -> <int> = 0 [7]
[1]  10 -> <int> = 1715558408065 [10]
[1]  11 -> <int> = 1715558385000 [11]
[1]  13 -> <int> = 1715558408065 [13]
[1]  14 -> <int> = 1220012 [14]
[1]  22 -> <int> = 0 [22]
[1]  26 -> <int> = 0 [26]
[1]  31 -> <int> = 1 [31]
[1]  [illegible] -> <int> = 1 [illegible]
[1]  45 -> <string> = "png" [45]
[1]  48 -> <string> = "1fd2d04e2c02fb26bbb16934c222f1ab" [48]
```



What about thumbnail recovery?

A recored will be added in the `item_properties` tabe with `key=thumbnail-cache-id-list` for the items that have their thumbnail cache. The cached thumbnails exists in the `thumbnails_cache` directory under the account profile.



DriveFS Sleuth

Automating the investigation

black hat
MIDDLE EAST AND AFRICA

DriveFS Sleuth!

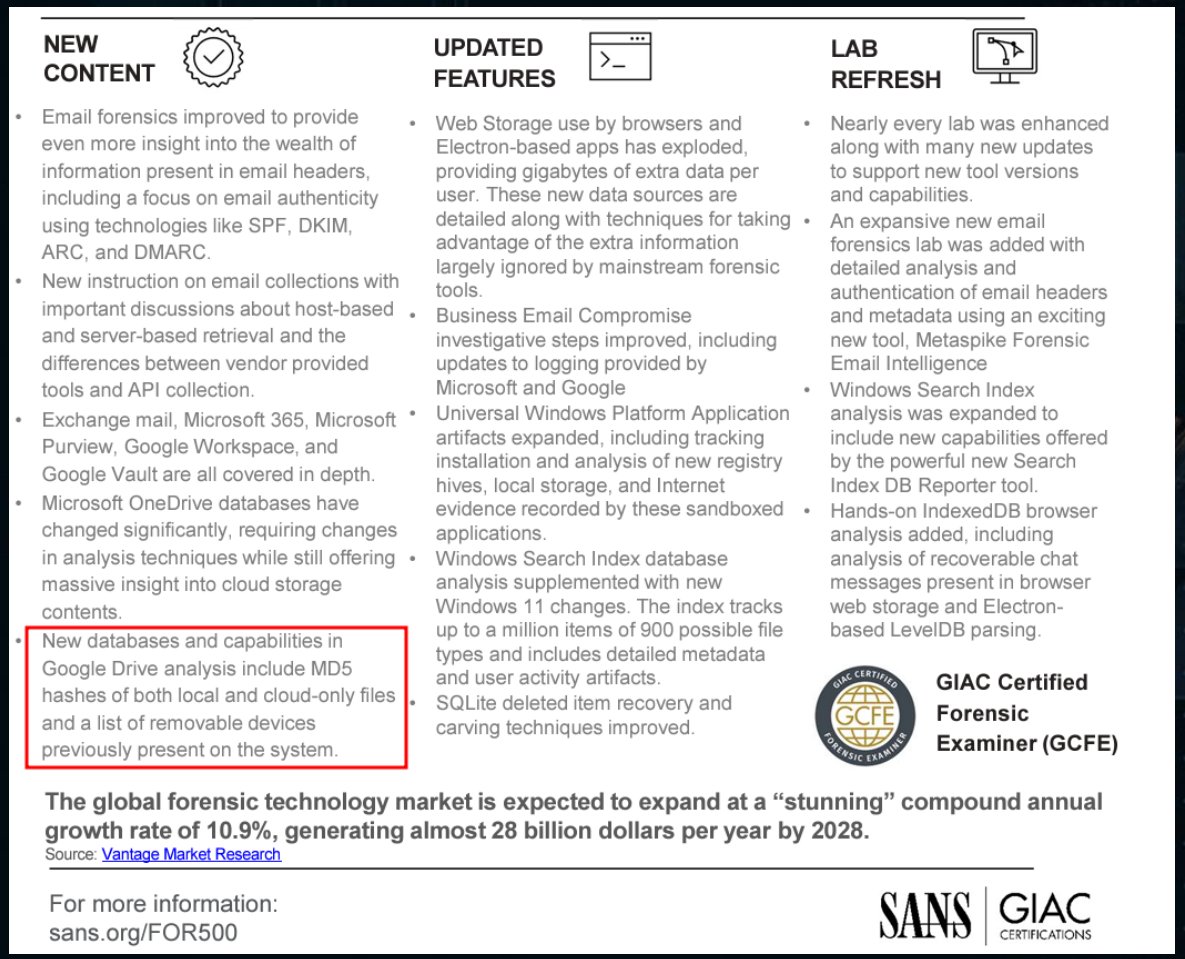
A python tool that automates the investigation of Google Drive (File System) disk artifacts.

https://github.com/AmgdGocha/DriveFS-Sleuth
https://pypi.org/project/drivefs-sleuth/

```
(drivefs_sleuth) drivefs_sleuth@drivefssleuth:~/drivefs_sleuth$ drivefs-sleuth --help
usage: DriveFS Sleuth [-h] -o OUTPUT [--accounts ACCOUNTS [ACCOUNTS ...]] [--regex REGEX [REGEX ...]] [-q QUERY_BY_NAME [QUERY_BY_NAME ...]] [--md5 MD5 [MD5 ...]] [--search-csv SEARCH_CSV] [--exact]
                      [--dont-list-sub-items] [--csv] [--html] [--recover-from-cache | --recover-search-results]
                      path

 ____  ____  _____     _____ ____  ____    ____  _     _____ _   _ _____ _   _
|  _ \|  _ \|_ _\ \   / | ____|  _ \/ ___|  / ___|| |   | ____| | | |_   _| | | |
| | | | |_) || | \ \ / /|  _| | |_) \___ \  \___ \| |   |  _| | | | | | | |_| |
| |_| |  _ < | |  \ V / | |___|  _ < ___) |  ___) | |___| |___| |_| | | | |  _  |
|____/|_| \_\___|  \_/  |_____|_| \_\____/  |____/|_____|_____|\___/  |_| |_| |_|

A tool for investigating Google Drive File Stream's disk forensic artifacts.

                                By: Amged Waqeh
                            Twitter: @amgdgocha
                        GitHub: https://github.com/AmgdGocha
                        Medium: https://medium.com/@amgedwageh
                    Linked In: https://www.linkedin.com/in/amgedwageh


positional arguments:
  path                  A path to the DriveFS folder. By default on a live system, it should exist in %LocalAppData%\Google\DriveFS.

options:
  -h, --help            show this help message and exit
  -o OUTPUT, --output OUTPUT
                        A path to a directory to save the output.
  --accounts ACCOUNTS [ACCOUNTS ...]
                        Specifies account id/s or emails separated by space to be processed, defaults to all the accounts.

Searching Arguments:
  --regex REGEX [REGEX ...]
                        Searches for files or folders by regular expressions. Multiple regex can be passed separated by spaces.
  -q QUERY_BY_NAME [QUERY_BY_NAME ...], --query-by-name QUERY_BY_NAME [QUERY_BY_NAME ...]
                        Searches for files or folders by name. The search will be case insensitive. Multiple file names can be passed separated by spaces.
  --md5 MD5 [MD5 ...]   Searches for files by the MD5 hash. Multiple hashes can be passed separated by spaces.
  --search-csv SEARCH_CSV
                        Searches for files or folders that satisfies the searching conditions in the provided CSV file.
  --exact               If selected, only files or folders with exact file names will be returned. The --query_by_name argument has to be passed. Defaults to False.
  --dont-list-sub-items
                        By default, if a folder matches the search criteria, the results will contain all of it's sub-items. This argument suppresses this feature to only return the folder without listing i
t's sub-items.

Output Formats:
  --csv                 Generates a CSV report. The CSV report will only contain information about the files and folders. Either --csv or --html should be specified.
  --html                Generates an HTML report. The HTML report contains comprehensive information about the analyzed artifacts.  Either --csv or --html should be specified.

Recovery Options:
  --recover-from-cache  Recover the cached items from the content cache.
  --recover-search-results
                        Recover the search results items that are cached.
```

The research and the tool have been added to the SANS FOR500 2024 update.

Ref: https://www.sans.org/blog/whats-new-in-for500-windows-forensic-analysis/



Demo Time!

Thanks!

DriveFS Sleuth

Your Ultimate Google Drive File Stream Investigator!

Amged Wageh

linkedin.com/in/amgedwageh/

amged_wageh@outlook.com