



26 - 28 NOVEMBER 2024
RIYADH, SAUDI ARABIA

DriveFS Sleuth

Your Ultimate Google Drive File Stream Investigator!

Presented BY: Amged Wageh



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



Amged Wageh

Incident Response Specialist - Kaspersky

Ex: Sr. DFIR Consultant - Cisco

DriveFS Sleuth Author



linkedin.com/in/amgedwageh/
amged_wageh@outlook.com



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



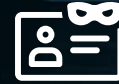
Frequent data leakage

A lot of classified documents were frequently leaked to Scribd!



Google Drive App. was in common

The suspected users had google drive for desktop application installed.



Hard to Investigate

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



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?

Deleted Files

Can we know about them?

Synced files context

What are the synced files?
What are their attributes?

Recovery

Can we recover interesting files from the cache?

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

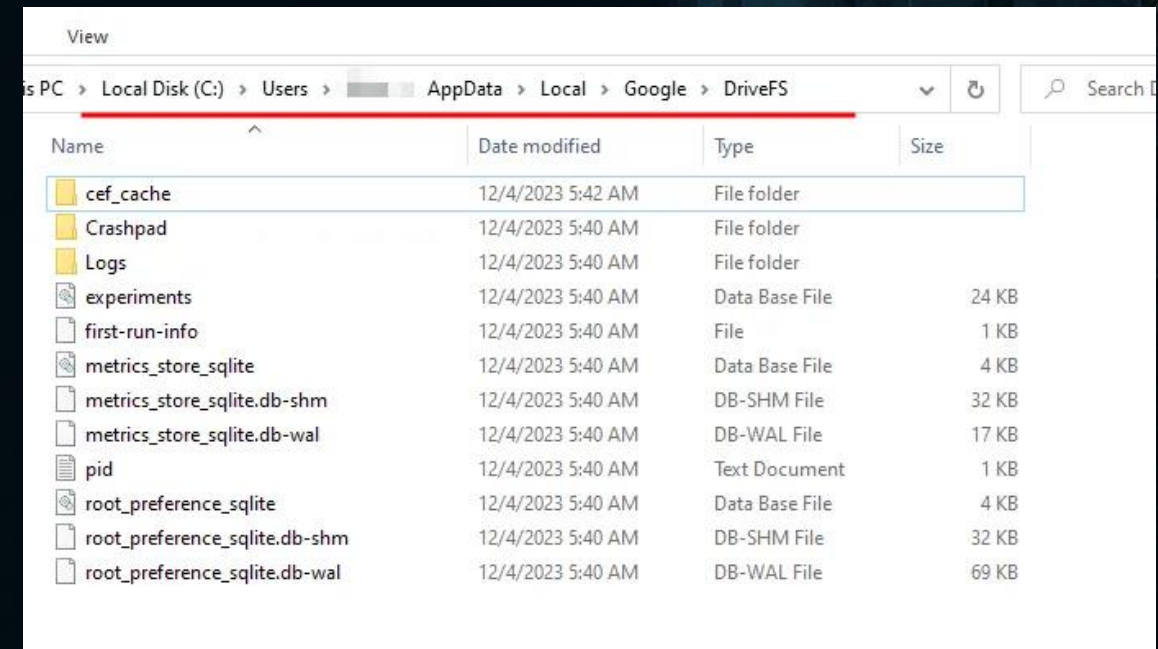
- Installation information
- CLSIDs for COM objects
- Shell Icon Overlays
- Shell Extensions
- Outlook Add-ons
- Persistence via Run

File system

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

What happens upon the application's first launch?

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

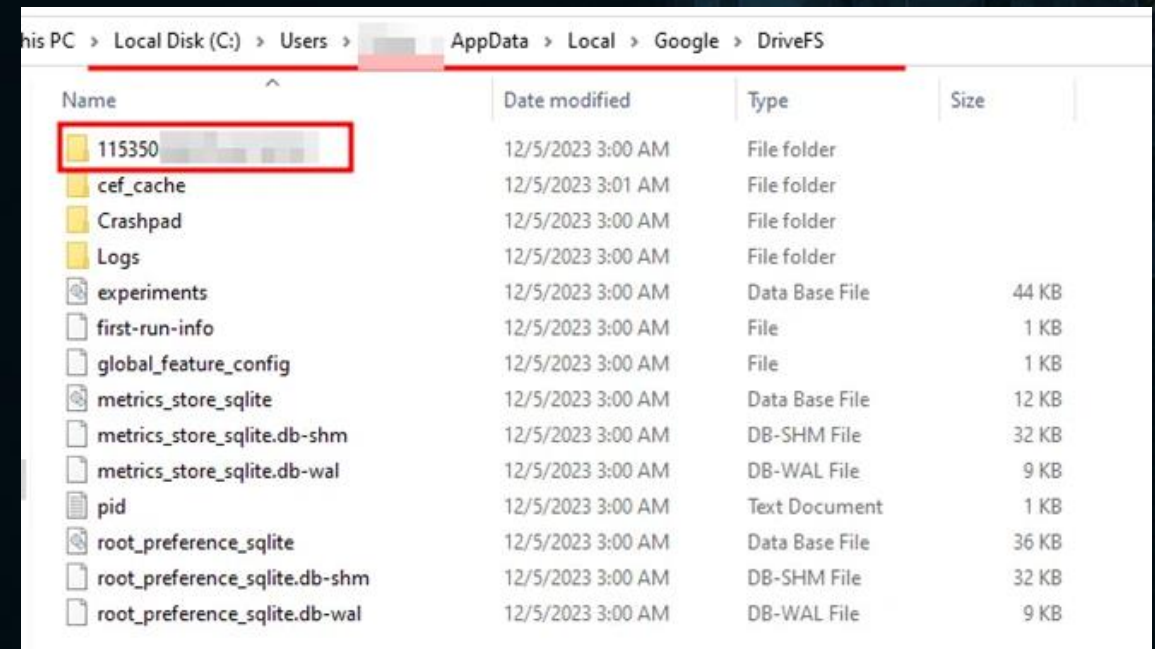


Name	Date modified	Type	Size
cef_cache	12/4/2023 5:42 AM	File folder	
Crashpad	12/4/2023 5:40 AM	File folder	
Logs	12/4/2023 5:40 AM	File folder	
experiments	12/4/2023 5:40 AM	Data Base File	24 KB
first-run-info	12/4/2023 5:40 AM	File	1 KB
metrics_store_sqlite	12/4/2023 5:40 AM	Data Base File	4 KB
metrics_store_sqlite.db-shm	12/4/2023 5:40 AM	DB-SHM File	32 KB
metrics_store_sqlite.db-wal	12/4/2023 5:40 AM	DB-WAL File	17 KB
pid	12/4/2023 5:40 AM	Text Document	1 KB
root_preference_sqlite	12/4/2023 5:40 AM	Data Base File	4 KB
root_preference_sqlite.db-shm	12/4/2023 5:40 AM	DB-SHM File	32 KB
root_preference_sqlite.db-wal	12/4/2023 5:40 AM	DB-WAL File	69 KB

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.*



Name	Date modified	Type	Size
115350	12/5/2023 3:00 AM	File folder	
cef_cache	12/5/2023 3:01 AM	File folder	
Crashpad	12/5/2023 3:00 AM	File folder	
Logs	12/5/2023 3:00 AM	File folder	
experiments	12/5/2023 3:00 AM	Data Base File	44 KB
first-run-info	12/5/2023 3:00 AM	File	1 KB
global_feature_config	12/5/2023 3:00 AM	File	1 KB
metrics_store_sqlite	12/5/2023 3:00 AM	Data Base File	12 KB
metrics_store_sqlite.db-shm	12/5/2023 3:00 AM	DB-SHM File	32 KB
metrics_store_sqlite.db-wal	12/5/2023 3:00 AM	DB-WAL File	9 KB
pid	12/5/2023 3:00 AM	Text Document	1 KB
root_preference_sqlite	12/5/2023 3:00 AM	Data Base File	36 KB
root_preference_sqlite.db-shm	12/5/2023 3:00 AM	DB-SHM File	32 KB
root_preference_sqlite.db-wal	12/5/2023 3:00 AM	DB-WAL File	9 KB

What is the nature of the artifacts?

Log files

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

SQLite Databases

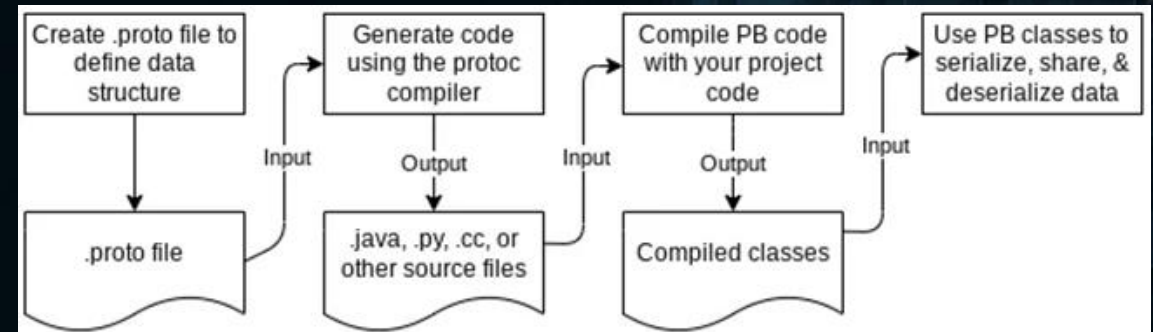
- Not encrypted.
- Stores a lot of enum values (needs experiments to figure them out).
- Depends on protobuf for storing important hidden info. (.proto files are not available)

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.

Research Findings

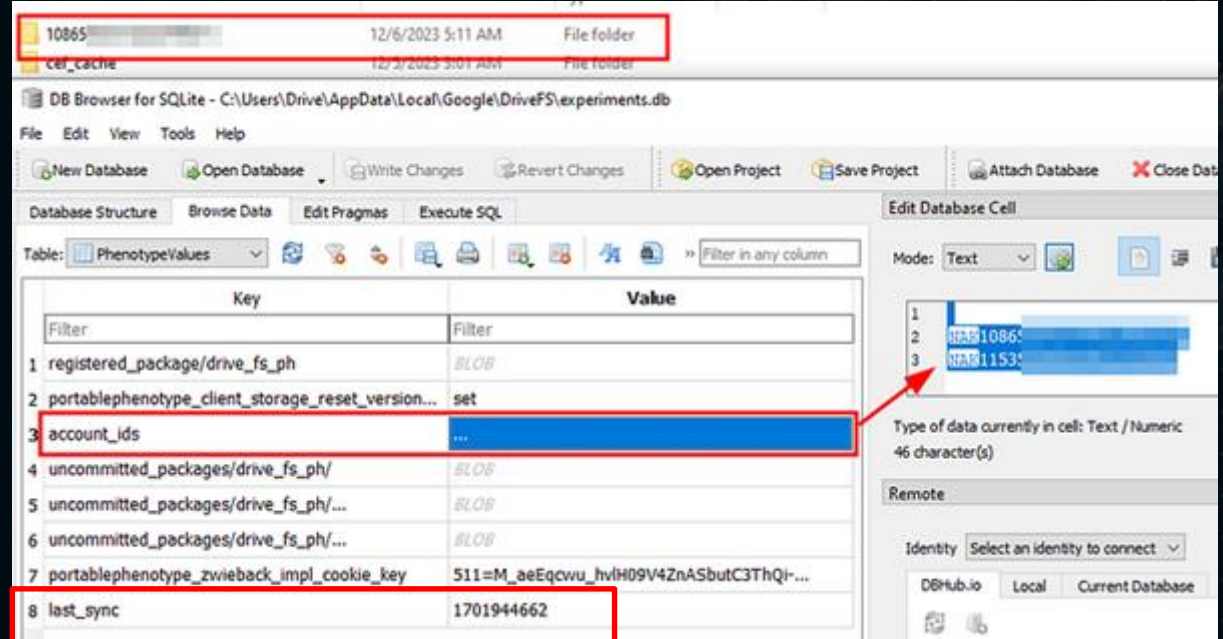
Installation Context

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.

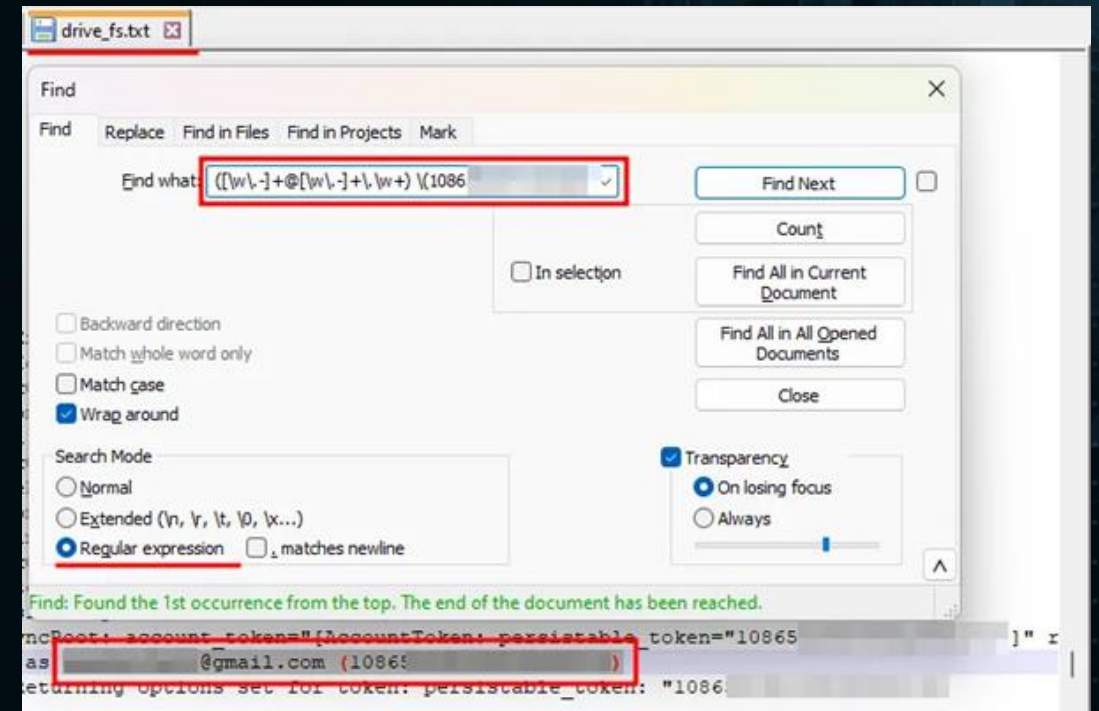


Who owns this ID?

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>\\)`



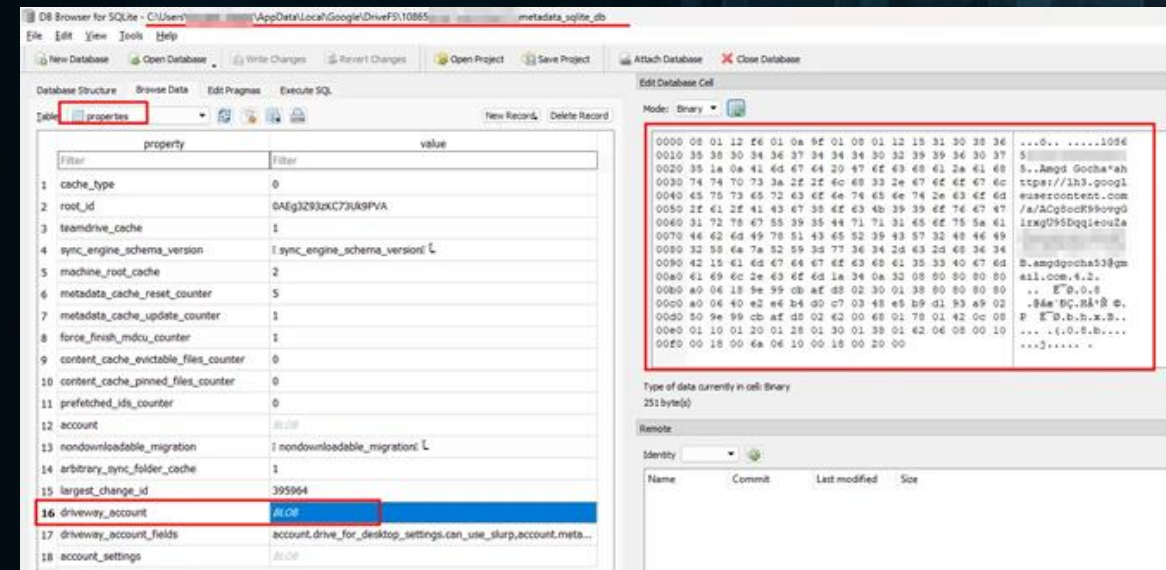
Some enrichment...

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.

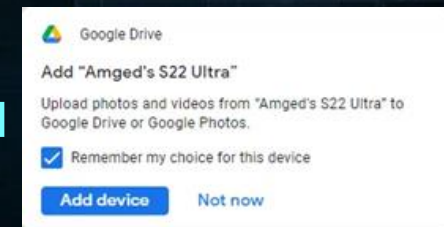
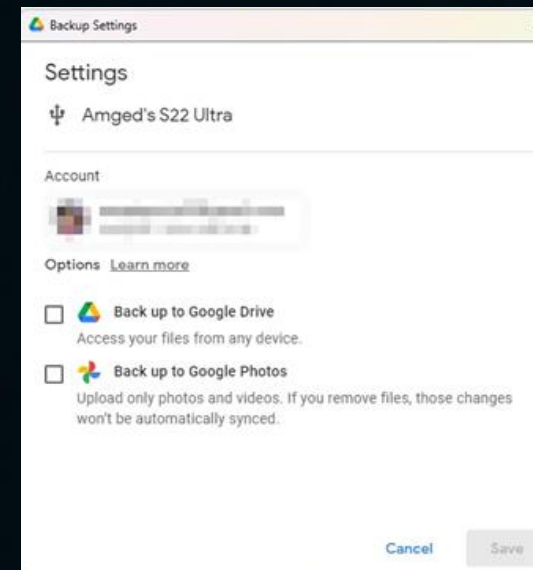


```
root:
1 <varint> = 1
2 <chunk> = message:
  1 <chunk> = message:
    1 <varint> = 1
    2 <chunk> = "10865"
    3 <chunk> = "Amgd Gocha"
    5 <chunk> = "https://lh3.googleusercontent.com/a/ACg8ocK99"
    8 <chunk> = "amgdgocha53@gmail.com"
  3 <chunk> = message:
```

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.

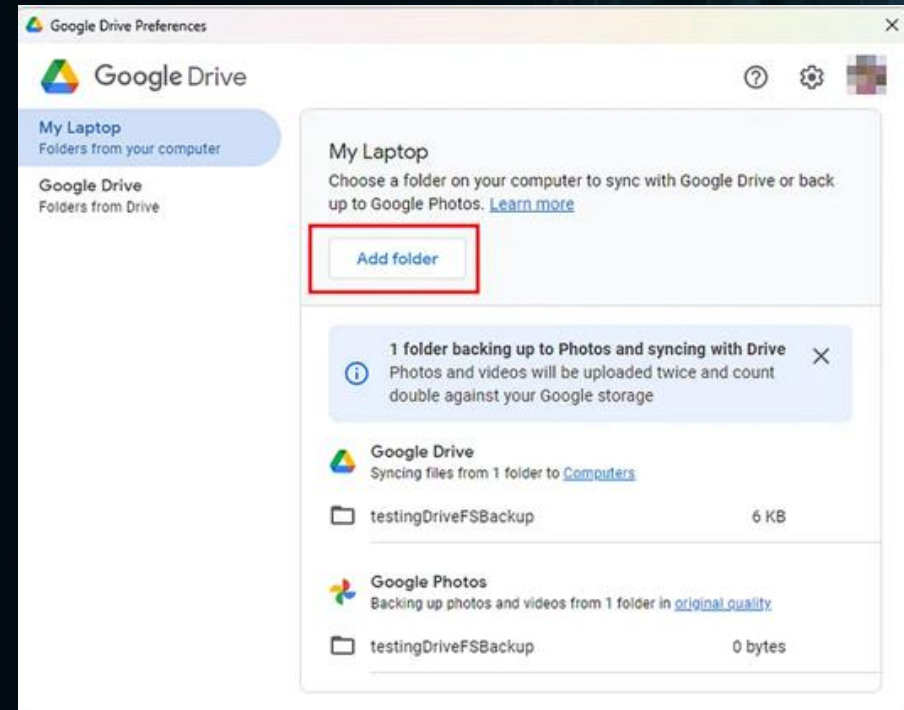
The media_id contains the same GUID stored in the registry.

	media_id	name	est_mount_point	fs_type	device_type	capacity	ignored
1	12ac0623-1907-4488-a59e-02335636db42	Windows	C:\	4	2	510720471040	0
2	2cbedcf-132f-416b-8b79-a1865fa1339f	Local Disk	D:\	4	2	1000068870144	0
3	2e7f874f-a7ef-445c-95ff-d1f4bd8473a8	NEW VOLUME	D:\	4	7	62107152384	1
4	ef54a46e-a523-46d7-bf6b-3af706e74bdf	My Passport - Amged	E:\	4	2	4000750497792	0
5	472601a7-194f-11ed-ad9b-2811a873702a	595.21.4	E:\	7	8	40860348416	0

Name
(Default)
{#00259b-7479-11ed-ad9b-2811a873702a}
{#0025a10-7479-11ed-ad9b-2811a873702a}
{177.VVolume\3e7a4c4a-6348-11ed-adb5-2811a873702a}
{177.VVolume\472601a7-194f-11ed-ad9b-2811a873702a}
{177.VVolume\472601a7-194f-11ed-ad9b-2811a873702a}
{177.VVolume\5708acc1-6160-11ee-adf0-2811a873702a}
{177.VVolume\ef1a3c-5500-11ed-adb5-2811a873702a}
{177.VVolume\b12f62c5-94f6-11ee-ae10-2811a873702a}
{177.VVolume\b958a293-32d2-11ee-adf0-2811a873702a}
{177.VVolume\b958a293-32d2-11ee-adf0-2811a873702a}
{177.VVolume\d9f139b-e7ab-11ed-add3-2811a873702a}
{\DosDevices\C:}
{\DosDevices\D:}
{\DosDevices\E:}
{\DosDevices\F:}
{\DosDevices\G:}

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.

root_id	metadata	media_id	title	root_path	account_token	sync_type	destination	medium	state	one_shot	is_my_drive	doc_id	seen_absolute
4	12ecd623-190...	12ecd623-190...	testingDriveF...	Users\Amged ...	10865804674...	1	1	1	2	0	0		C:\Users\Amg...
7	12ecd623-190...	12ecd623-190...	testingDriveF...	Users\Amged ...	10865804674...	2	2	1	2	0	0		C:\Users\Amg...
13	\\7\USB\681d...		Amged's S22 ...		10865804674...	2	1	2	2	1	0		

id_type	value
1 max_root_id	8

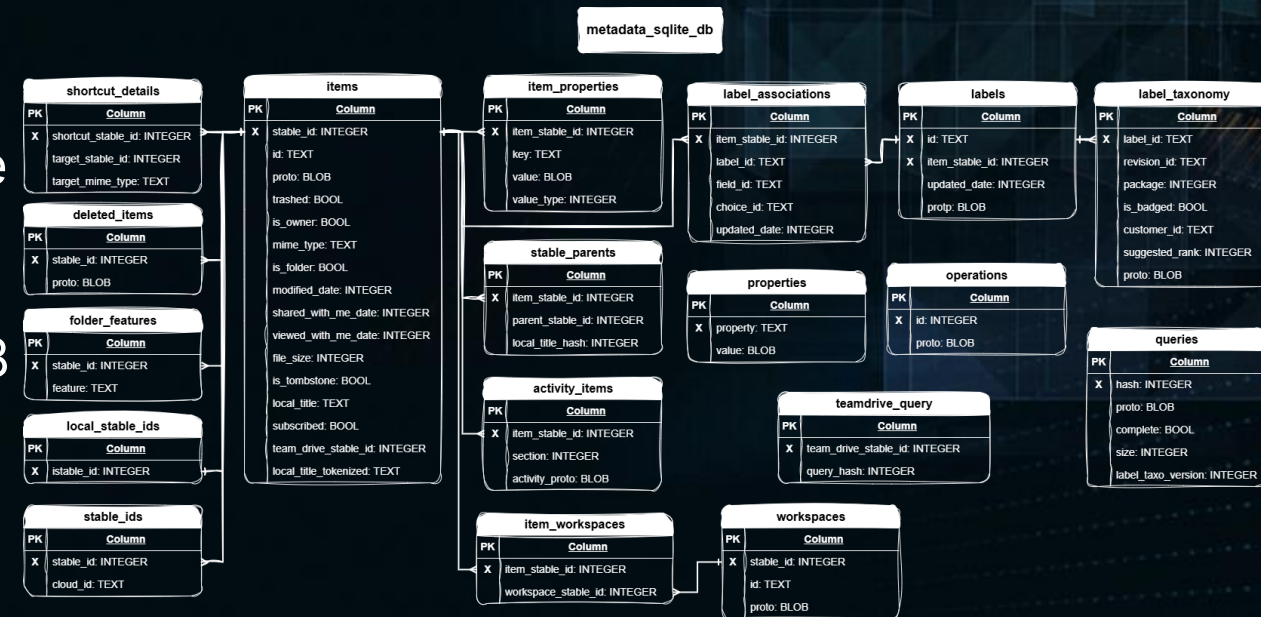
The one_shot column stores 1 if the user didn't select the "Remember my choice for this device." option and the record will be deleted once unplugged, and 0 otherwise.

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



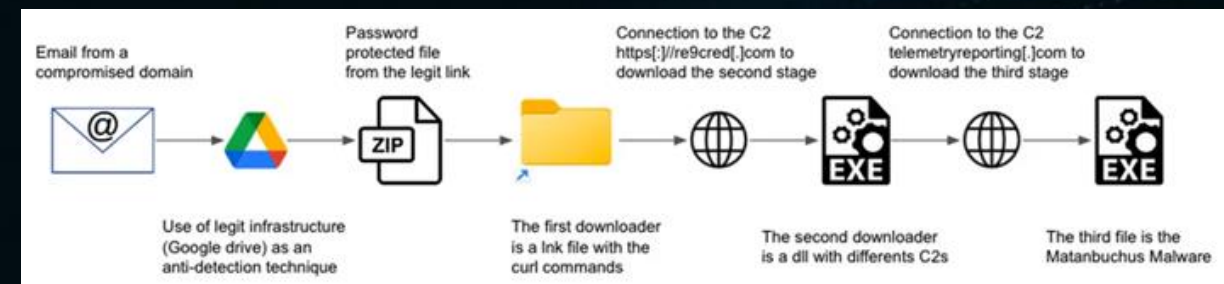
It exists under the user's profile
%LocalAppData%\Google\DriveFS\

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.

stable_id	id	proto	trashed	is_owner	mime_type	is_folder	modified_date	red_with_me_d	rved_by_me_da	file_size	onbst	local_title
112353	1md5f8hml...	CZz	0	1	application/vn...	1	1702205463196	0	1702205463196	0	0	DriveFS-Research



Matanbuchus malware attack flow

Ref: <https://intelligence.abnormalsecurity.com/blog/google-drive-matanbuchus-malware>

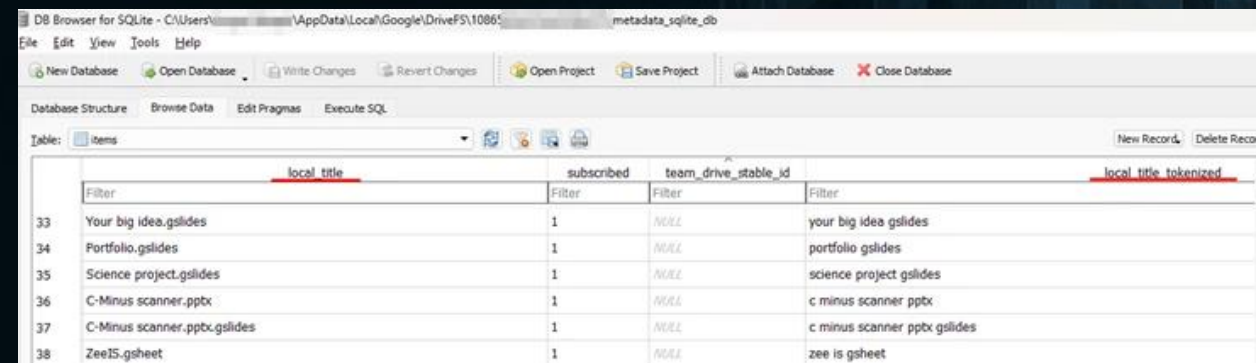
What are the items info.?

local_title and local_title_tokenized
for the item name.

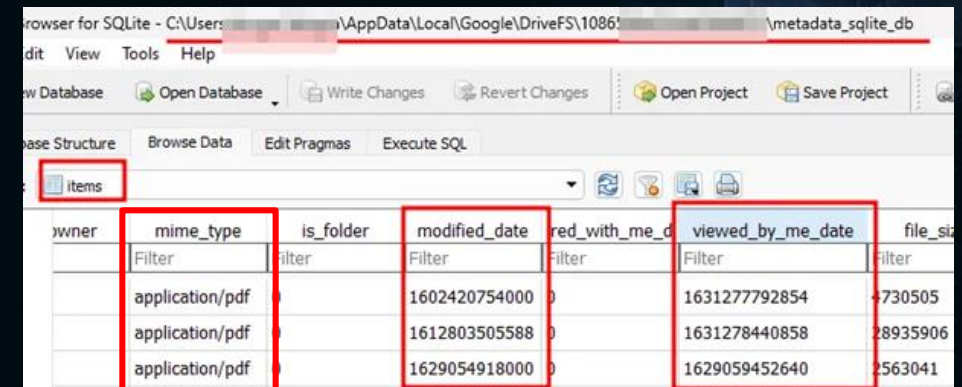
modified_date for the last modification
time

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

mime_type of the item.



	local_title	subscribed	team_drive_stable_id	local_title_tokenized
33	Your big idea.gslides	1	NULL	your big idea gslides
34	Portfolio.gslides	1	NULL	portfolio gslides
35	Science project.gslides	1	NULL	science project gslides
36	C-Minus scanner.pptx	1	NULL	c minus scanner pptx
37	C-Minus scanner.pptx.gslides	1	NULL	c minus scanner pptx gslides
38	ZeeIS.gsheet	1	NULL	zee is gsheet

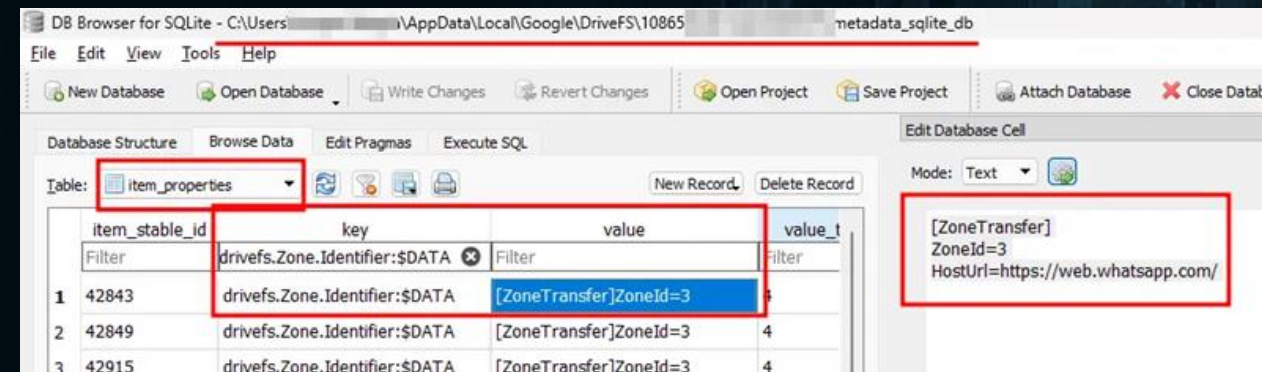


owner	mime_type	is_folder	modified_date	red_with_me_d	viewed_by_me_date	file_size
	application/pdf		1602420754000		1631277792854	730505
	application/pdf		1612803505588		1631278440858	8935906
	application/pdf		1629054918000		1629059452640	563041

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!



DB Browser for SQLite - C:\Users\... \AppData\Local\Google\DriveFS\10865 metadata_sqlite_db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: item_properties

	item_stable_id	key	value	value_t
		drivefs.Zone.Identifier:\$DATA	Filter	filter
1	42843	drivefs.Zone.Identifier:\$DATA	[ZoneTransfer]ZoneId=3	
2	42849	drivefs.Zone.Identifier:\$DATA	[ZoneTransfer]ZoneId=3	4
3	42915	drivefs.Zone.Identifier:\$DATA	[ZoneTransfer]ZoneId=3	4

Mode: Text

[ZoneTransfer]
ZoneId=3
HostUrl=https://web.whatsapp.com/

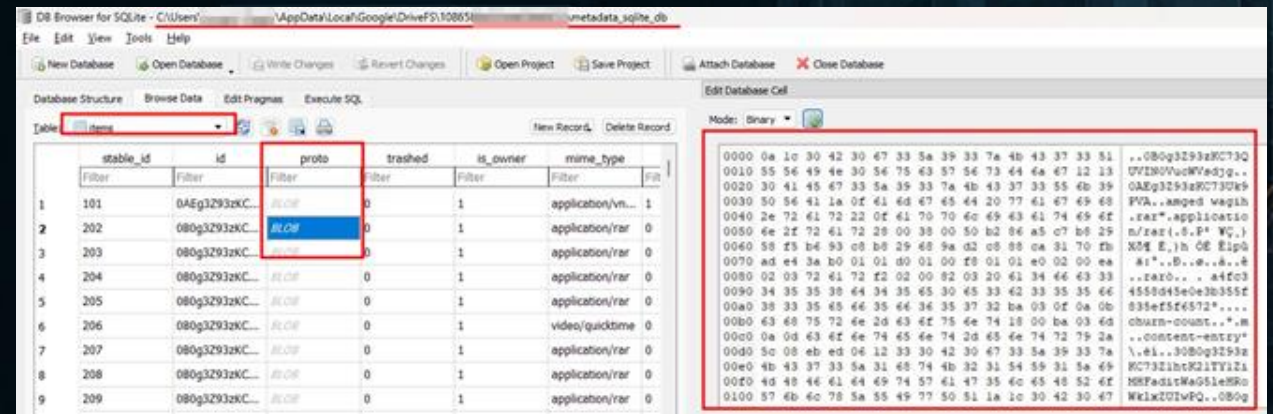
Revealing hidden info.

– 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!



stable_id	id	proto	trashed	is_owner	mime_type
1	101	0AEg3Z93zK...	0	1	application/vn...
2	202	080g3Z93zK...	0	1	application/rar
3	203	080g3Z93zK...	0	1	application/rar
4	204	080g3Z93zK...	0	1	application/rar
5	205	080g3Z93zK...	0	1	application/rar
6	206	080g3Z93zK...	0	1	video/quicktime
7	207	080g3Z93zK...	0	1	application/rar
8	208	080g3Z93zK...	0	1	application/rar
9	209	080g3Z93zK...	0	1	application/rar

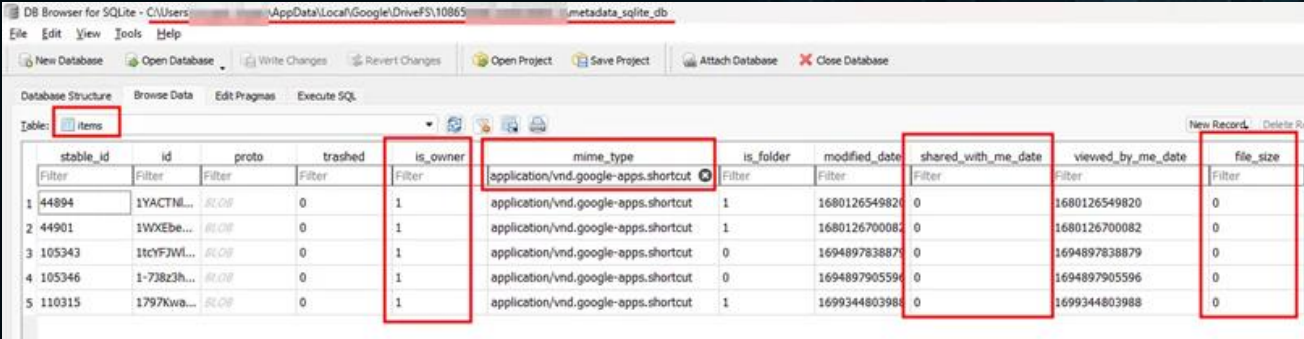
```
root:
1 <chunk> = "080g3Z93zK..."
2 <chunk> = "0AEg3Z93zK..."
3 <chunk> = "amged wagih.rar"
4 <chunk> = "application/rar"
5 <varint> = 0
7 <varint> = 0
10 <varint> = 1423931163442
11 <varint> = 1423932971893
13 <varint> = 1703509371162
14 <varint> = 123279099
22 <varint> = 1
26 <varint> = 0
31 <varint> = 1
44 <varint> = 0
45 <chunk> = "rar"
46 <chunk> = "empty_chunk"
48 <chunk> = "a4fc34558d45e0e3b355f835ef5f6572" MD5
55 <chunk> = message:
  1 <varint> = "churn-count"
  3 <varint> = 0
56 <chunk> = message:
```


What are the shortcut files?

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

mime_type is application/vnd.google-apps.shortcut.

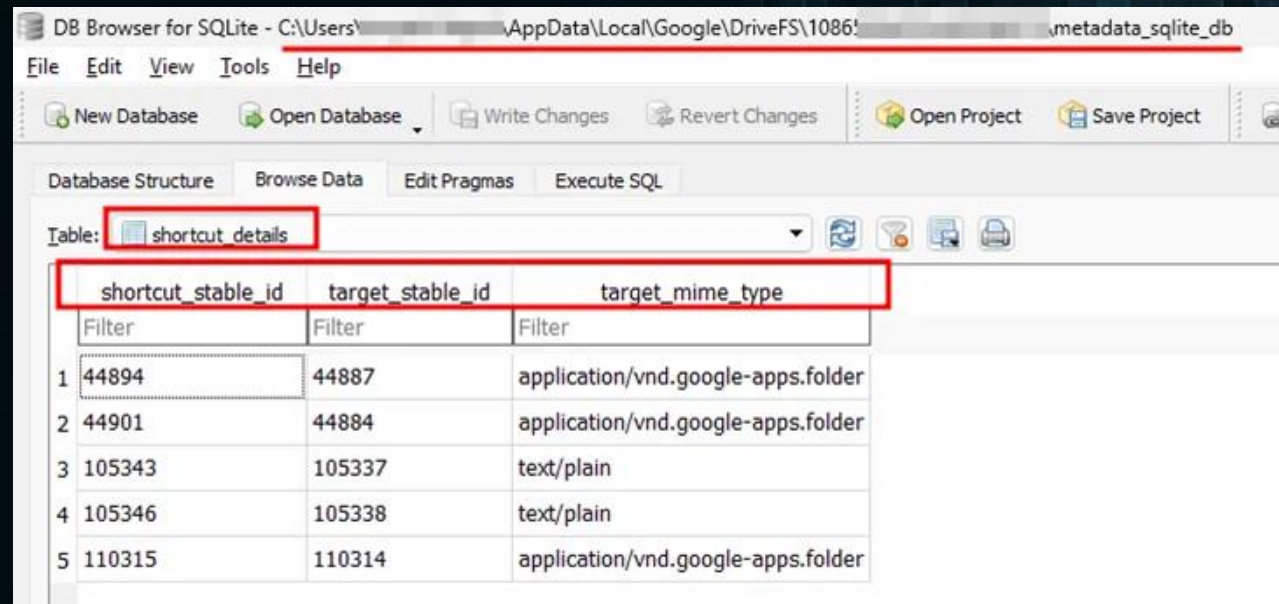
File_size will always be 0.



stable_id	id	proto	trashed	is_owner	mime_type	is_folder	modified_date	shared_with_me_date	viewed_by_me_date	file_size
1	44894	1YACTN...	0	1	application/vnd.google-apps.shortcut	1	1680126549820	0	1680126549820	0
2	44901	1WKEbe...	0	1	application/vnd.google-apps.shortcut	1	1680126700082	0	1680126700082	0
3	105343	1tcYFJWL...	0	1	application/vnd.google-apps.shortcut	0	1694897838879	0	1694897838879	0
4	105346	1-738z3h...	0	1	application/vnd.google-apps.shortcut	0	1694897905596	0	1694897905596	0
5	110315	1797Kva...	0	1	application/vnd.google-apps.shortcut	1	1699344803988	0	1699344803988	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.



The screenshot shows the DB Browser for SQLite application. The title bar indicates the database path is C:\Users\...\.AppData\Local\Google\DriveFS\1086\...metadata_sqlite_db. The 'Table:' dropdown is set to 'shortcut_details'. The table structure is displayed with three columns: shortcut_stable_id, target_stable_id, and target_mime_type. The table contains five rows of data, each with an index number in the first column.

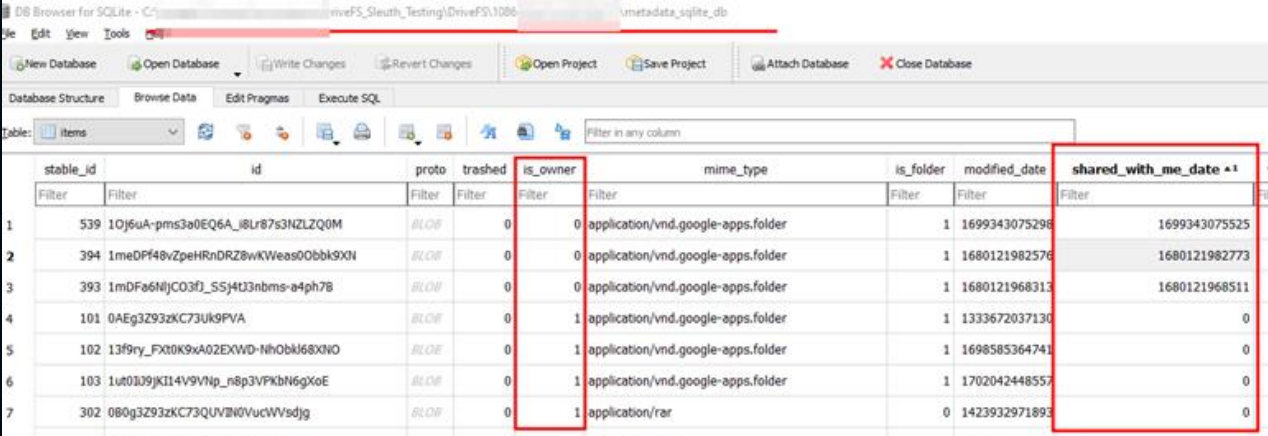
	shortcut_stable_id	target_stable_id	target_mime_type
1	44894	44887	application/vnd.google-apps.folder
2	44901	44884	application/vnd.google-apps.folder
3	105343	105337	text/plain
4	105346	105338	text/plain
5	110315	110314	application/vnd.google-apps.folder

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.



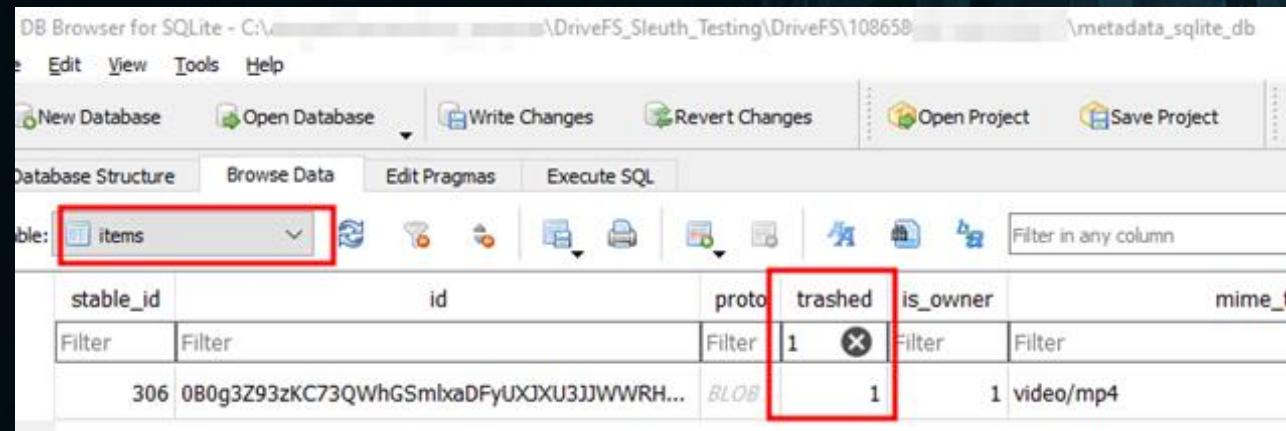
stable_id	id	proto	trashed	is_owner	mime_type	is_folder	modified_date	shared_with_me_date
1	539 10j6uA-pms3a0EQ6A_j8Lr87s3NZLZQ0M	BLOB	0	0	application/vnd.google-apps.folder	1	1699343075298	1699343075525
2	394 1meDPf48vZpeHRnDRZ8wKVweas0Obbk9XN	BLOB	0	0	application/vnd.google-apps.folder	1	1680121982576	1680121982773
3	393 1mDFa6tHjCO3fj_SSj4t3nbms-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-NhObkI68XVO	BLOB	0	1	application/vnd.google-apps.folder	1	1698585364741	0
6	103 1u0t0J9JKI14V9Vnq_n8p3VPkbN6XoE	BLOB	0	1	application/vnd.google-apps.folder	1	1702042448557	0
7	302 080g3Z93zKC73QUVWVucVWsdjg	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.



DB Browser for SQLite - C:\Users\... \DriveFS_Sleuth_Testing\DriveFS\108658\... \metadata_sqlite_db

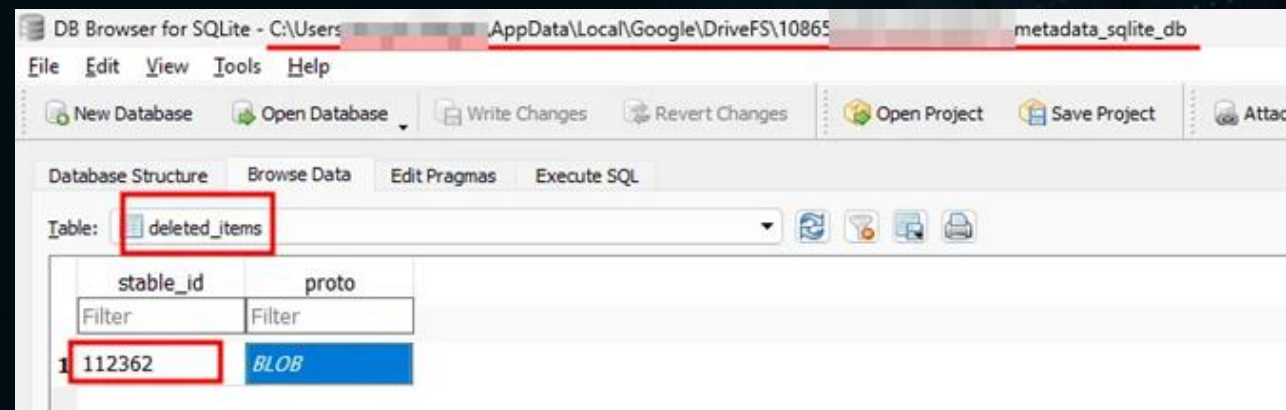
File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragas Execute SQL

Table: items

stable_id	id	proto	trashed	is_owner	mime
Filter	Filter	Filter	1	Filter	Filter
306	0B0g3Z93zKC73QWhGSmbxaDFyUXJXU3JJWWRH...	BLOB	1	1	video/mp4



DB Browser for SQLite - C:\Users\... \AppData\Local\Google\DriveFS\108658\... metadata_sqlite_db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach

Database Structure Browse Data Edit Pragas Execute SQL

Table: deleted_items

stable_id	proto
Filter	Filter
112362	BLOB

Absolutely! by decoding the protobuf message.

```

root:
1 <chunk> = "lv39cd6WiMhYEXyuzW0sf1cU5ZsHcq6TX"
2 <chunk> = "1md5f4hmD7Xljd38wiGpbsrLEwMSUCzZ"
3 <chunk> = "history"
4 <chunk> = "application/octet-stream"
5 <varint> = 0
6 <varint> = 1
7 <varint> = 1702209341555
8 <varint> = 1693392584702
9 <varint> = 1702209341555
10 <varint> = 3637248
11 <varint> = 0
12 <varint> = 1
13 <varint> = 1
14 <varint> = 0
15 <chunk> = empty chunk
16 <chunk> = empty chunk
17 <chunk> = "2ab55cafe262b0ef630598134a58ac42"
18 <chunk> = message:
19 | <chunk> = "churn-count"
20 | <varint> = 0
21 <chunk> = message:
22 | <chunk> = "content-entry"
23 | <chunk> = message:
24 | | <varint> = 112361
25 | | <chunk> = "0B0g3Z93zKC73VjIcy95Q0tGYUtOLJjQ3HFu#ubEttV3LzPQ"
26 | | <chunk> = "lv39cd6WiMhYEXyuzW0sf1cU5ZsHcq6TX"
27 | | <varint> = 3637248
28 <chunk> = message:
29 | <chunk> = "drives.Zone.Identifier"
30 | <chunk> = bytes (26)
31 | 0000 5B 5A 6F 6E 65 54 72 61 6E 73 66 65 72 50 00 0A 5A 6F 6E 65 49 64 30 33 [ZoneTransfer]. ZoneId=3
32 | 0010 00 0A
33 <chunk> = message:
34 | <chunk> = "drives.Zone.Identifier:$DATA"
35 | <chunk> = "[ZoneTransfer]|\r\nZoneId=3|\r\nReferrerUrl=https://www.google.com/search?q=wait=true&r\n"
36 <chunk> = message:
37 | <chunk> = "local-cache-reason"
38 | <varint> = 1
39 <chunk> = message:
40 | <chunk> = "local-content-modified-date"
41 | <varint> = 1693392584702
42 <chunk> = message:
43 | <chunk> = "local-title"
44 | <chunk> = "history"

```

```
55 <chunk> = message:
  1 <chunk> = "trashed-locally-metadata"
  5 <chunk> = bytes (102)
    0000 02 00 00 00 00 00 00 00 00 80 37 00 00 00 00 00 F0 34 1B EA 5F 2B DA 01 .....7.....4...+...
    0018 25 00 00 00 47 00 3A 00 5C 00 4D 00 79 00 20 00 44 00 72 00 69 00 76 00 %...G.:\.M.y. Driv
    0030 65 00 5C 00 44 00 72 00 69 00 76 00 65 00 46 00 53 00 2D 00 52 00 65 00 e\.Drive\F.S.-Re
    0048 73 00 65 00 61 00 72 00 63 00 68 00 5C 00 68 00 69 00 73 00 74 00 6F 00 searc.h\h.isto
    0060 72 00 79 00 00 00 r.y...
55 <chunk> = message:
  1 <chunk> = "trashed-locally-name"
  4 <chunk> = "$RMKGJPN"
55 <chunk> = message:
  1 <chunk> = "version-counter"
  3 <varint> = 22
57 <chunk> = "1693392584702""
63 <varint> = 1
69 <varint> = 1702209350325
76 <varint> = 2
78 <chunk> = "080g3Z93zKc73Vjlicy9500tGYUt0L3JjQ3hFUmRubEttV3lzPQ"
```

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

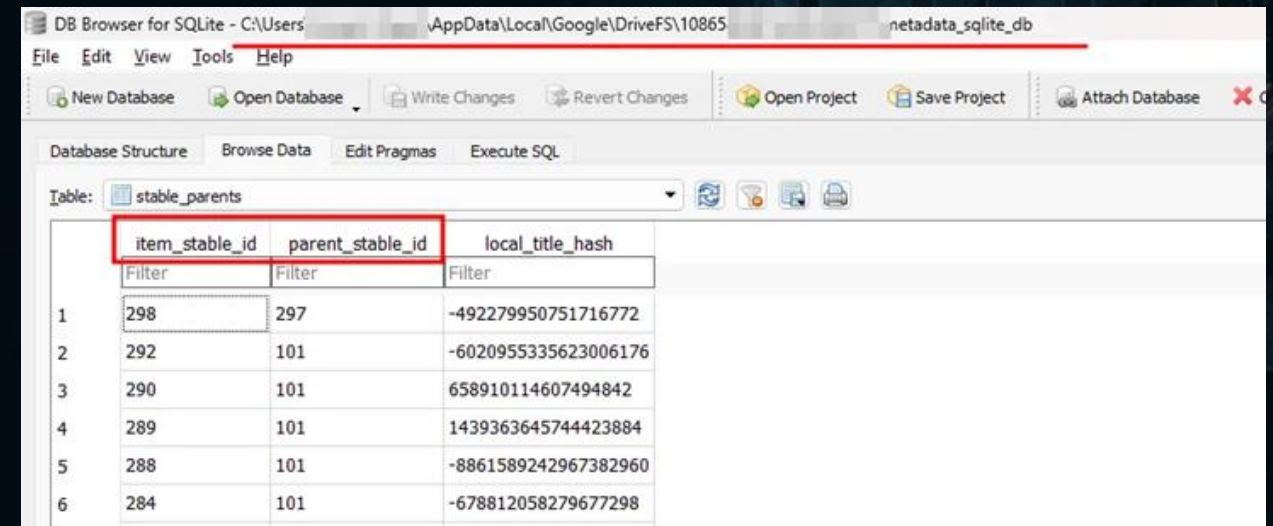
Revealing the proto scheme...

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).



The screenshot shows the DB Browser for SQLite application. The 'Table' dropdown is set to 'stable_parents'. The table structure is displayed with three columns: 'item_stable_id', 'parent_stable_id', and 'local_title_hash'. The first two columns are highlighted with a red box. Below the table structure, a list of data rows is shown, each with a row number and the values for the three columns.

	item_stable_id	parent_stable_id	local_title_hash
1	298	297	-492279950751716772
2	292	101	-6020955335623006176
3	290	101	658910114607494842
4	289	101	1439363645744423884
5	288	101	-8861589242967382960
6	284	101	-678812058279677298

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.

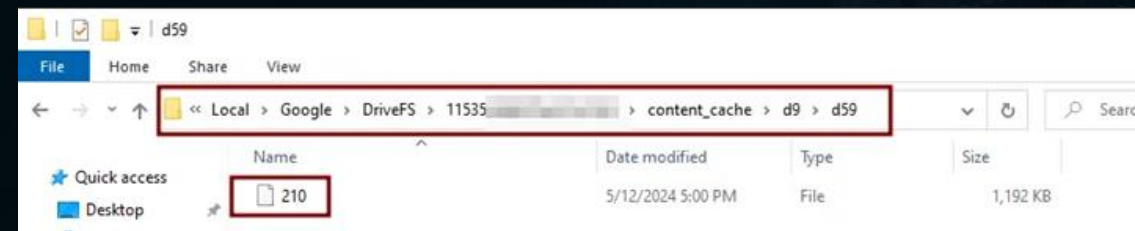
That id is the filename under the content_cache directory.

item_stable_id	key	value	value_type
206	Filter	Filter	Filter
1	206 churn-count	0	2
2	206 content-entry	AKO8	2
3	206 local-cache-reason	1	2
4	206 local-title	screen-shot-2018-07-11-at-5-06-35...	3
5	206 modified-date	1715558432215	2



```
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> = "11XkchaDZ0b9xQn8WFUTSJvvfjZqVRbNt" [3]
[1] 4 -> <int> = 1220012 [4]
```

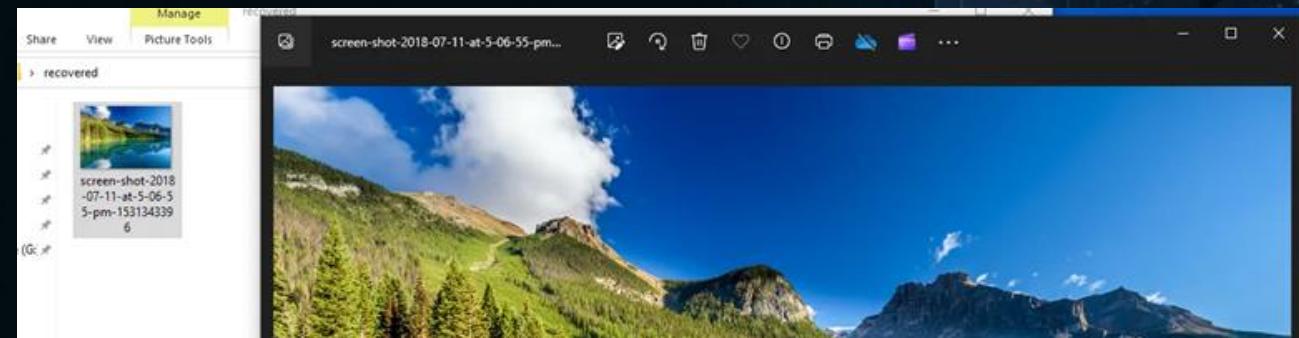


Recovered item renaming...

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

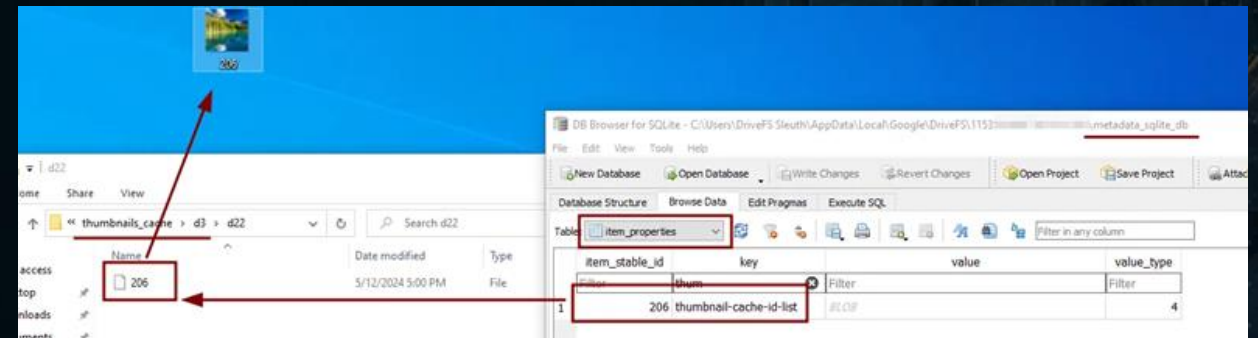
```
PS C:\Users\DriveFS Sleuth> protodeep -t protobuf 'C:\users\DriveFS Sleuth\Desktop\bins\206-proto.bin'
Finding the protobuf starting chunk...
[+] Bruteforce index : 0

[1] 1 -> <string> = "11XkchaDZ0b9xQn8WFUTSJvvfjZqVRbNr" [1]
[1] 2 -> <string> = "1-Ry4MCidEv_cFFh-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] 44 -> <int> = 1 [44]
[1] 45 -> <string> = "png" [45]
[1] 48 -> <string> = "1fd2d04e2c02fb26bbb16934c222f1ab" [48]
```



What about thumbnail recovery?

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



DriveFS Sleuth

Automating the investigation

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@drivefsleuth:~/drivefs_sleuth$ drivefs-sleuth --help
usage: DriveFS Sleuth [-h] -o OUTPUT [--accounts ACCOUNTS [ACCOUNTS ...]] [--regex REGEX [REGEX ...]] [--q QUERY_BY_NAME [QUERY_BY_NAME ...]] [--mfs MFS [MFS ...]] [--search-csv SEARCH_CSV] [--exact]
                        [--dont-list-sub-items] [--csv] [--html] [--recover-from-cache] [--recover-search-results]
                        path

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

By: Amgd Wageh
Twitter: @amgdgocha
GitHub: https://github.com/AmgdGocha
Medium: https://medium.com/@amgdgocha
Linked In: https://www.linkedin.com/in/amgdgocha

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.
  --mfs MFS [MFS ...]
                        Searches for files by the MFS 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/>

NEW CONTENT



- Email forensics improved to provide even more insight into the wealth of information present in email headers, including a focus on email authenticity using technologies like SPF, DKIM, ARC, and DMARC.
- New instruction on email collections with important discussions about host-based and server-based retrieval and the differences between vendor provided tools and API collection.
- Exchange mail, Microsoft 365, Microsoft Purview, Google Workspace, and Google Vault are all covered in depth.
- Microsoft OneDrive databases have changed significantly, requiring changes in analysis techniques while still offering massive insight into cloud storage contents.
- New databases and capabilities in Google Drive analysis include MD5 hashes of both local and cloud-only files and a list of removable devices previously present on the system.

UPDATED FEATURES



- Web Storage use by browsers and Electron-based apps has exploded, providing gigabytes of extra data per user. These new data sources are detailed along with techniques for taking advantage of the extra information largely ignored by mainstream forensic tools.
- Business Email Compromise investigative steps improved, including updates to logging provided by Microsoft and Google
- Universal Windows Platform Application artifacts expanded, including tracking installation and analysis of new registry hives, local storage, and Internet evidence recorded by these sandboxed applications.
- Windows Search Index database analysis supplemented with new Windows 11 changes. The index tracks up to a million items of 900 possible file types and includes detailed metadata and user activity artifacts.
- SQLite deleted item recovery and carving techniques improved.

LAB REFRESH



- Nearly every lab was enhanced along with many new updates to support new tool versions and capabilities.
- An expansive new email forensics lab was added with detailed analysis and authentication of email headers and metadata using an exciting new tool, Metaspoke Forensic Email Intelligence
- Windows Search Index analysis was expanded to include new capabilities offered by the powerful new Search Index DB Reporter tool.
- Hands-on IndexedDB browser analysis added, including analysis of recoverable chat messages present in browser web storage and Electron-based LevelDB parsing.




**GIAC Certified
Forensic
Examiner (GCFE)**

The global forensic technology market is expected to expand at a “stunning” compound annual growth rate of 10.9%, generating almost 28 billion dollars per year by 2028.

Source: [Vantage Market Research](#)

For more information:
sans.org/FOR500

SANS | **GIAC**
CERTIFICATIONS



Demo
Time!

Thanks!

DriveFS Sleuth

Your Ultimate Google Drive File Stream
Investigator!

Amged Wageh



[linkedin.com/in/amgedwageh/](https://www.linkedin.com/in/amgedwageh/)
amged_wageh@outlook.com