Step Action 8.4 (Android Extraction)

Part I – Locate Google Maps Data

Step 1 – Launch Physical Analyzer and open the Android extraction image.

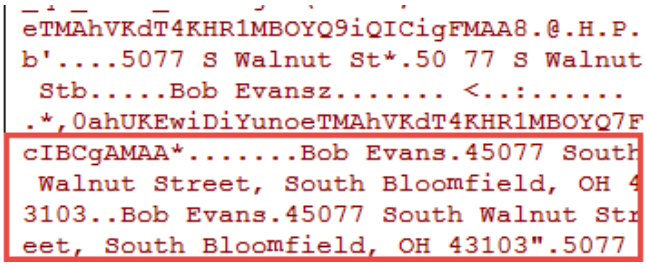
Step 2 – Go to **Analyzed Data**, and view **Installed Applications.**

Step 3 – Locate the "**Maps**" application and view the installed path.

Step 4 – Navigate to **Userdata/data/com.google.android.apps.maps** and open the databases folder.

Step 5 – Open the databases folder and view the "**gmm_storage.db**" file.

Step 6 – View the **Hex View** tab and observe you can see various addresses embedded in the file.



Step 7 – Switch to the **Database View** tab. Select "**gmm_storage_table**" and observe the "**_data**" column contains BLOB data.

Step 8 – Right click on "**gmm_storage.db**" and select **Save As**. Save the file to your scripts folder.

Step 9 – Minimize Physical Analyzer. Launch **SQLite Browser** and drag the "**gmm_storage.db**" file into it to open the file.

Step 10 – Switch to the **Browse Data** tab and select the **gmm_storage** table. Double click on the BLOB date in the _data column, line number 3.

Step 11 – Scroll down to line **0x02 C0**. View the address and destination for "**Palomino Club**". You will need to change the viewer pull down menu to "Binary"

Note: While the data can be exported here, as we learned in Module 2, there are 52 rows of data to export, one line at a time. This is a time-consuming

process, especially if there are many more rows of data. We will use a script to export the data for us.

Part II – Export BLOB Data

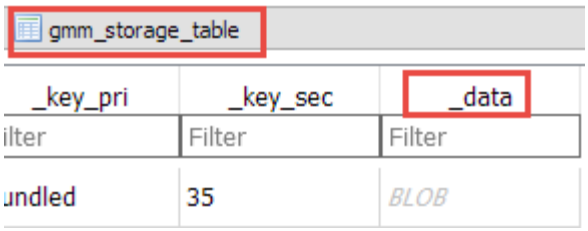
Step 1 – Locate the script in your Scripts folder named "**Google_Maps_BLOB.py**". This script uses Python 3.

Step 2 – Right click on the script and open it with **Notepad++.**

Step 3 – View the comment on line 12. This script will run on any SQLite table containing BLOB data. You only have to change the table name and column name which contains the BLOB data.

Step 4 – View line 14, "**TABLE_NAME**". This is where the name of the TABLE we want the data from is entered. Note the name is already populated.

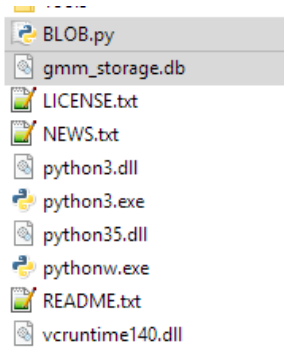
Step 5 – View line 16, "**FIELD_NAME**". This is the title of the column which contains the BLOB data. Note this name is already populated.



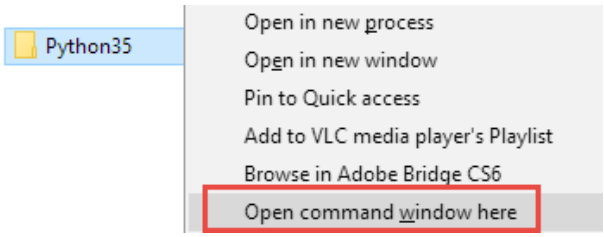

```
12  # PICK TABLES/FIELDS TO WORK WITH
13
14  TABLE_NAME = "gmm_storage_table"
15
16  FIELD_NAME = "_data"
```

Step 6 – Close **Notepad++,** saving any changes if needed.

Step 7 -- Copy the **BLOB.py** file and the **gmm_storage.db** files to the installation folder for **Python 3**. This step isn't necessary if Python 3 is properly reflected in your PATH statement. If you have doubts or errors running the script, copy it to the Python 3 installation folder.



Step 8 – Right click on your Python 3 installation folder, while holding down the "Shift" key and select "**Open command window here**".

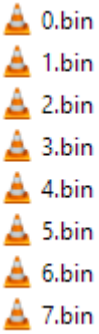


Step 9 – At the command line type "**python.exe Google_maps_BLOB.py gmm_storage.db**" without quotes.

```
>python.exe Google_maps_BLOB.py gmm_storage.db
```

Step 10 – After a pause the cursor will return at the command prompt. Close the DOS window.

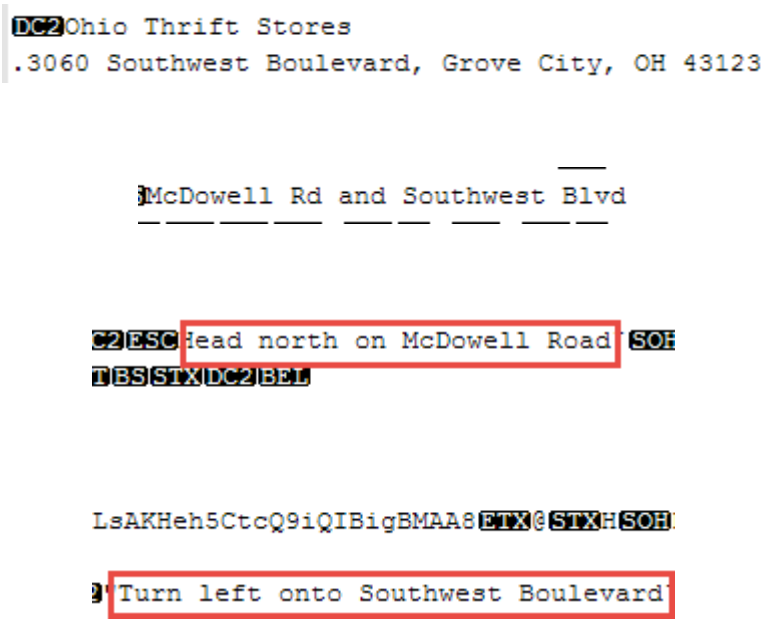
Step 11 – Navigate to the Python 3 folder on your computer, or the location where the script ran. Observe the exported data.



Step 12 – Note the BIN extension is due to the data containing binary data. Each of these files can be opened with Notepad++.

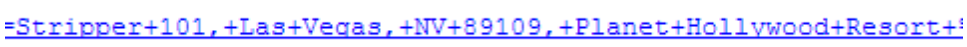
Part III – View the exported data

Step 1 – Right click on the file "**33.bin**" and select **Open with Notepad++**. View the location and directions to this navigated to location.



Step 2 – Close Notepad++, right click on the file "**9.bin**" and open it in **Notepad++.**

Step 3 – View the address and note the lack of directions. This is data which was entered into the search bar for the Google Maps application, and the returned data.



These files could be copied into your case folder, and reviewed in a text program or forensic tool.

Step 4 – (**OPTIONAL**) Shift-Right click on the folder containing your extracted BLOB files (Python 3 if you haven't moved them) and **open a command prompt**.

Step 5 – At the command line type "**copy /b *.bin data.txt**" and press enter.

This command will copy all the binary data (/b) from every file which has a .bin extension (*.bin) into a file named "data.txt" (data.txt).

Step 6 – Right click and open the new data.txt file in Notepad++ to view the results.

Part IV – View Location and Direction Data

Step 1 – Return to the Samsung GSM SM_G900A Galaxy S5 image in Physical Analyzer.

Step 2 – Navigate to the **userdata/Root/data/com.google.android.apps.maps/cache/http** folder.

Step 3 – Locate the file **9d81b79604f9ded39b8f954590799c17.0**

NOTE: The files which have a .0 extension contain direction information. The files with a .1 extension will sometimes contain a graphic image file of locations on the route. Note that the files ending in .1 don't necessarily represent the intended destination.

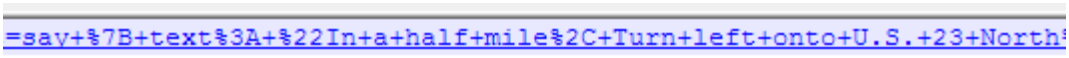
Step 4 – Double click the file you located above to open it

The data is contained in binary format, which isn't in a format to be read within Physical Analyzer.

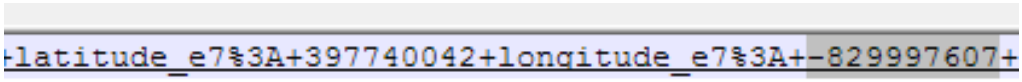
Step 5 – Right click in the file and select **Save As**. Save the file to the **Case Folder** on the **Desktop**.

Step 6 – Minimize Physical Analyzer and right click on the **9d81b79604f9ded39b8f954590799c17.0** file and Open it in **Notepad ++.**

Step 7 – View the direction data given to the device user after the "**say**" prompt.



Step 8 – Continue viewing on the same line to locate the **latitude** and **longitude** data. This is where the user was when the directions were given.

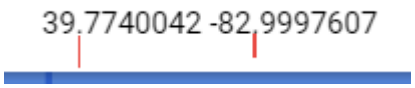


Step 9 – Open a web browser and navigate to Google Maps

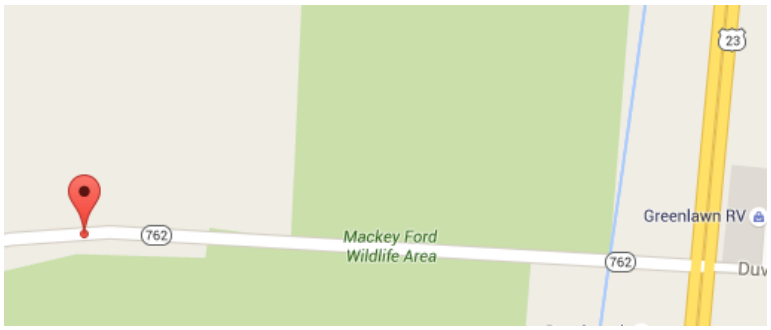
Step 10 – Enter the latitude and longitude coordinates, keeping care to use the appropriate integer from the data.

Latitude: +397740042
Longitude: -829997607

Step 11 – Place a decimal point in the appropriate place in each coordinate. Note the appropriate integer for the longitude.

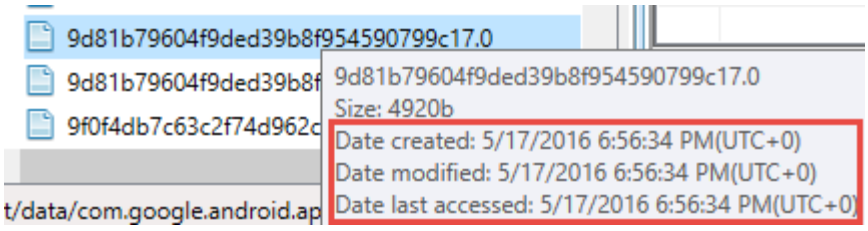


Step 12 – Press enter and view the location of the user when the directions were given.



Step 13 – Return to Physical Analyzer.

Step 14 – Highlight the file and view its **Created** time as reported by the file system. This is where the user was located during that file creation.



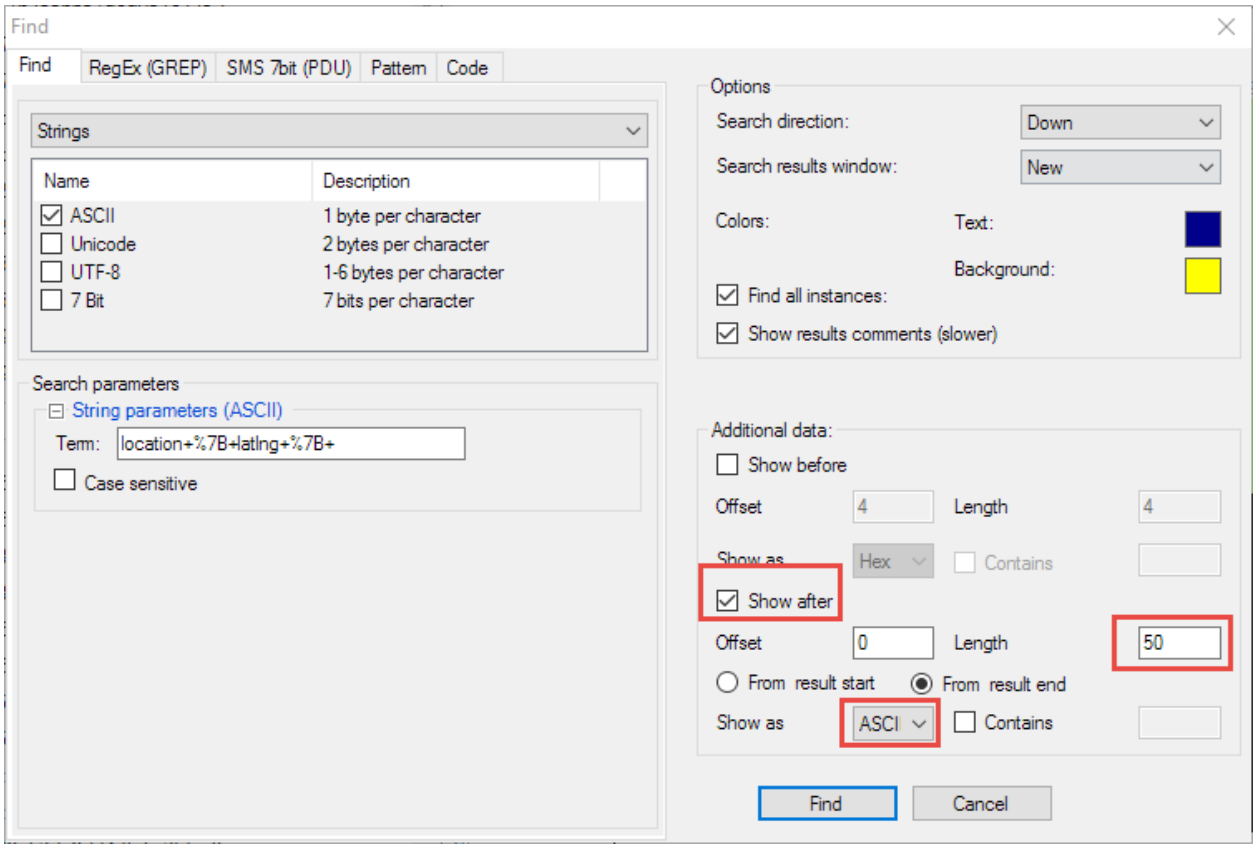
Step 15 – In the **Project Tree**, open the **memory image**.

Step 16 – Using **Find**, choose **Strings** and **ASCII.**

Step 17 – In the search field enter "**location+%7B+latlng+%7B+**".

This value is present in each of these files which contains latitude and longitude coordinates.

Step 18 – In the **Additional data** section, check the box for **Show After**. Change the Length to **50**, **From result end**, and change Show as to **ASCII**. Press **Find**.



Step 19 – View the results.

Step 20 – Click the Export to Excel button at the top and save the file to your desktop.

Source	Additional after
/Root/data/com.google.android.apps.maps/files/SavedClientParameters.data	latitude_e7%3A+$LAT_E7+longitude_e7%3A+$LONG_E7+%
/Root/data/com.google.android.apps.maps/cache/http/3bf93713d9659a1ca6bede72aa726580.0	latitude_e7%3A+398790028+longitude_e7%3A+-83065002
/Root/data/com.google.android.apps.maps/cache/http/aa36c1bff1b0efd69e5d83e351d51d88.0	latitude_e7%3A+397428833+longitude_e7%3A+-83025010
/Root/data/com.google.android.apps.maps/cache/http/9d81b79604f9ded39b8f954590799c17.0	latitude_e7%3A+397740042+longitude_e7%3A+-82999760
/Root/data/com.google.android.apps.maps/cache/http/4688d611b00d9afa7b44089dfa1b2922.0	latitude_e7%3A+397428833+longitude_e7%3A+-83025010
/Root/data/com.google.android.apps.maps/cache/http/a2936d0b810536e05844ecb20677ef18.0	latitude_e7%3A+397428771+longitude_e7%3A+-83024769

From here Excel queries can be run to change the coordinates to the proper format. Note the source filename is displayed as well to obtain Created time information from Physical Analyzer.