#OSDFCon
OPEN SOURCE DIGITAL FORENSICS CONFERENCE

Analyzing Apps and Communications with Autopsy

Raman Arora
Danny Smyda

Goal

- Introduce and review Communications Analysis features in Autopsy.
- Introduce new module writing support for apps.
- Get feedback on additional apps you'd like support for.

Why Use Autopsy for Apps and Communications

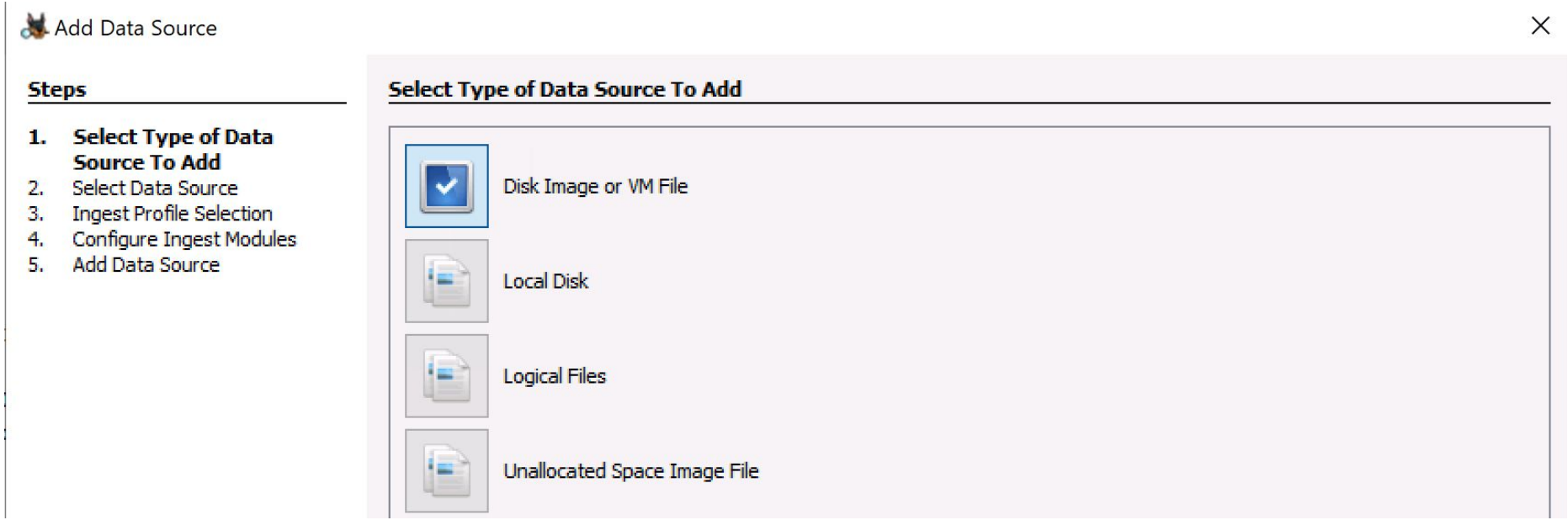
- Dedicated communications interface allows you to quickly focus on relevant accounts and messages.
- Support for both computer and phone formats allow you to see and correlate all data in a single case.
- Plug-in framework allows you and others to write modules to support new apps.

Supported Inputs

- Autopsy does not acquire data from a phone.

- Supported Inputs:
  - Physical images
    - File systems: HFS+, Ext4, Yaffs2, FAT (media card)
  - File system dumps
  - USB-attached device

Adding a Physical Image



Adding a Physical Image

Add Data Source

Steps

1. Select Type of Data Source To Add
2. **Select Data Source**
3. Ingest Profile Selection
4. Configure Ingest Modules
5. Add Data Source

Select Data Source

Path:

c:\case_inputs\case123\android_image.bin

Browse

☐ Ignore orphan files in FAT file systems

Time zone: (GMT-5:00) America/New_York

Sector size: Auto Detect

Hash Values (optional):

MD5:

SHA-1:

Data Parsed from Computer Media

Emails

- PST
- MBOX
- EML

Contacts

- VCards

Browsers

- Chrome
- Firefox
- IE
- Edge
- Safari

Underlined items are new since last year.

Data Parsed from Phone Media

Messaging/Calling

- Android SMS, Call Logs
- Words With Friends
- Tango
- WhatsApp
- Skype
- Facebook Messenger
- Viber
- Line
- TextNow
- IMO

File Sharing

- ShareIt
- Xender
- Zapya

Browsers

- Android
- Opera
- S(amsung)Browser

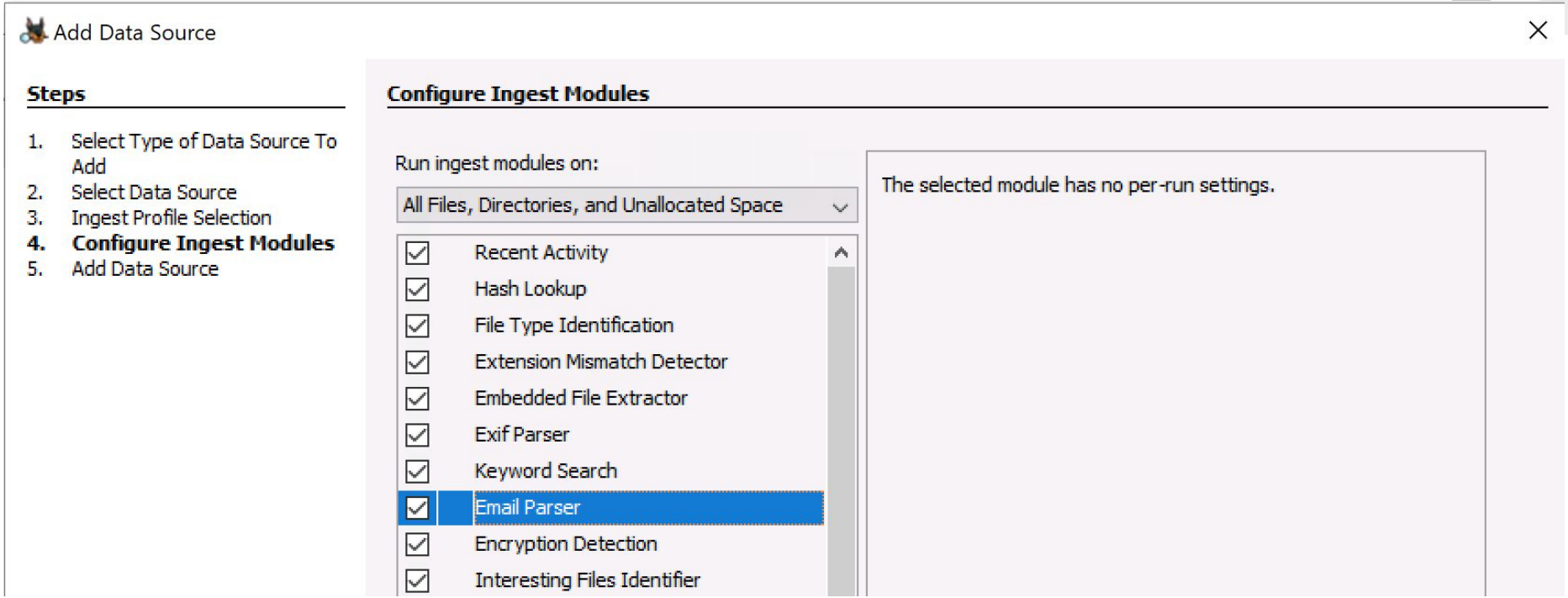
Maps

- Orux
- Google Maps

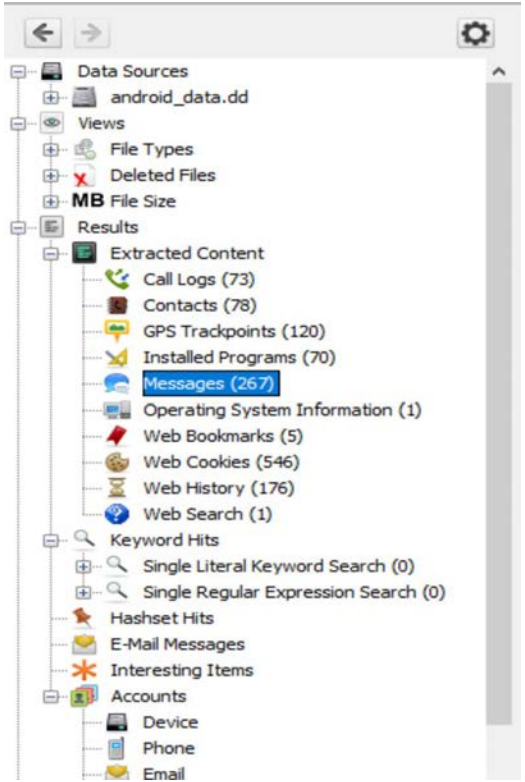
Underlined are new.

Many more to come...

Select Ingest Modules



Viewing Results in Tree



- Generic display
- Organized by artifact type
- No filtering or sorting

View From Tree

viber_messages	Outgoing	+16784357227	2019-09-27 11:51:24 EDT	You wanna hang out tonight?	Viber Messenger	1
viber_messages	Incoming	+17812314569	2019-09-27 11:52:04 EDT	Going out with girlfriend. Tomorrow?	Viber Messenger	1
viber_messages	Outgoing	+16784357227	2019-09-27 14:37:54 EDT	Hey Darlene, wanna go to movies tonight?	Viber Messenger	1
viber_messages	Incoming	+17812314569	2019-09-27 14:38:20 EDT	Sure. You have one in mind?	Viber Messenger	1
viber_messages	Outgoing	+16784357227	2019-09-27 14:39:20 EDT	How about Avengers? Popcorn on me!	Viber Messenger	1
viber_messages	Incoming	+17812314569	2019-09-27 14:39:50 EDT	See you there at 7	Viber Messenger	1

- Generic table display
- Columns are Name/values
- No filtering, some sorting

Communications Viewer - Overview

An intuitive and user-friendly interface to view communications.

- Organizes accounts that were found (such as phone number or email).
- Shows all messages, calls, and contact book entries associated with an account.
- Allows for filtering based on account types and dates.

Funded by DHS S&T

Communications Viewer

Communications Visualization - Editor

Communications Visualization

Account Types:

- Device
- Email
- Facebook
- IMO
- LINE
- Phone
- ShareIt
- Skype
- TextNow

Uncheck All | Check All

Devices:

- android_data.dd

Uncheck All | Check All

Date Range (America/New_York):

Start: December 31, 2013

End: September 30, 2019

Browse | Visualize

Account	Device	Type	Items
23afe152-1984-4a0a-bcee-1e	android_data.dd	Device	201
+16783002112	android_data.dd	Viber	28
u8a7e68b3365c07e64fade456	android_data.dd	LINE	26
live:.cid.2d133a28af8d285f	android_data.dd	Skype	25
+16174534409	android_data.dd	TextNow	13
+17742780342	android_data.dd	Phone	12
8:live:.cid.ab01b1b1545b16d9	android_data.dd	Skype	12
8:live:.cid.8f278e2e9be3ae57	android_data.dd	Skype	12
+17812314569	android_data.dd	Viber	11
100039669064643	android_data.dd	Facebook	11
2000107721184083	android_data.dd	IMO	10
ua9cf19436f475243938b1001	android_data.dd	LINE	9
ub830497e95d7a2adf0610ff2	android_data.dd	LINE	9
2000222005372359	android_data.dd	IMO	8
uaa130a5b1246ea6106f576cc	android_data.dd	LINE	6
live:btforensics1	android_data.dd	Skype	6
100000829961026	android_data.dd	Facebook	6
+15086697112	android_data.dd	Phone	5
+16179441839	android_data.dd	TextNow	5
+15086235961	android_data.dd	TextNow	5
+15084359878	android_data.dd	Viber	5
+14123976868	android_data.dd	Viber	5
+12029784687	android_data.dd	Viber	5
100039666335138	android_data.dd	Facebook	4

Summary | Messages | Call Logs | Contacts | Media Attachments

Account Contacts

Book Entries: 7

Communication References: 0

File References in Current Case

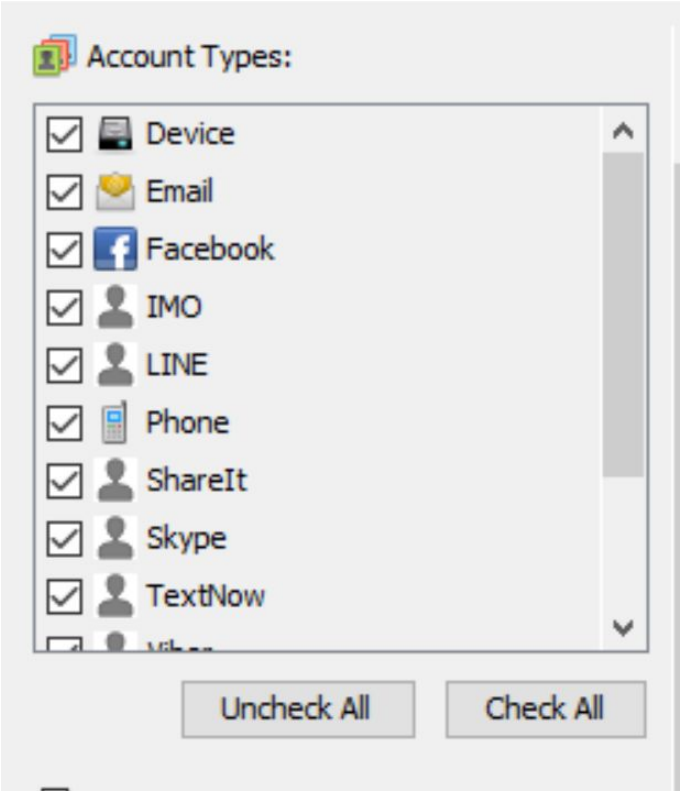
Path

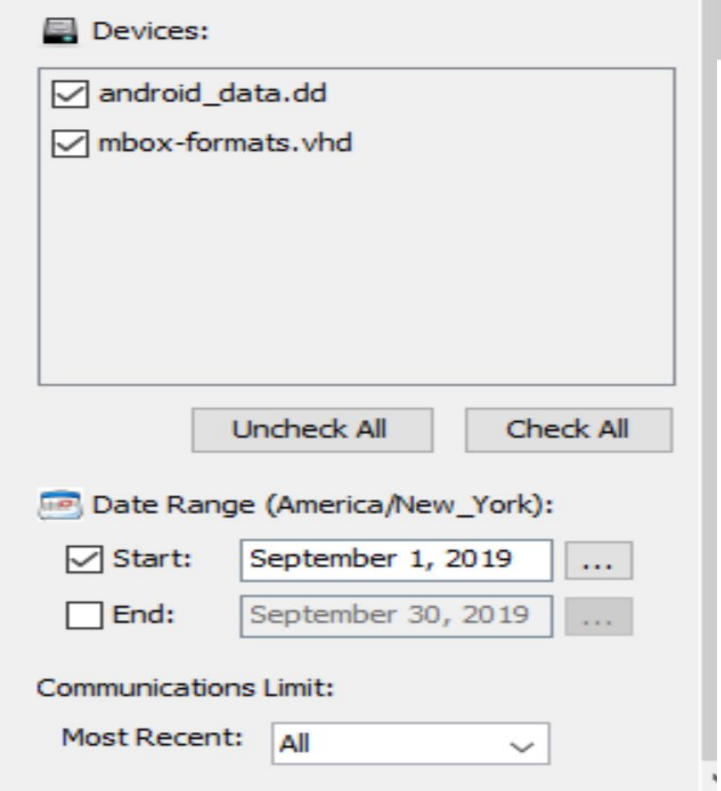
- /img_android_data.dd/data/com.android.providers.contacts
- /img_android_data.dd/data/com.android.providers.telepho
- /img_android_data.dd/data/com.whatsapp/databases/wa.c
- /img_android_data.dd/data/com.viber.voip/databases/vibe
- /img_android_data.dd/data/com.viber.voip/databases/vibe

Other Occurrences

<Enable Central Respository to see Other Occurrences>

Communications Viewer - Filtering





Communications Viewer - Accounts Browser

Browse | Visualize

Account	Device	Type	Items
1b397bcf-1455-4c0d-8736-29fe	android_data.dd	Device	201
+16784357227	android_data.dd	Viber	28
u8a7e68b3365c07e64fade456	android_data.dd	LINE	26
live:.cid.2d133a28af8d285f	android_data.dd	Skype	25
+16174534409	android_data.dd	TextNow	13
8:live:.cid.ab01b1b1545b16d9	android_data.dd	Skype	12
8:live:.cid.8f278e2e9be3ae57	android_data.dd	Skype	12
+17812314569	android_data.dd	Viber	11
100039669064643	android_data.dd	Facebook	11
2000107721184083	android_data.dd	IMO	10
ua9cf19436f475243938b1001e	android_data.dd	LINE	9
ub830497e95d7a2adf0610ff2a	android_data.dd	LINE	9
2000222005372359	android_data.dd	IMO	8
uaa130a5b1246ea6106f576cc4	android_data.dd	LINE	6
live:btforensics1	android_data.dd	Skype	6
100000829961026	android_data.dd	Facebook	6
+16179441839	android_data.dd	TextNow	5
+15086235961	android_data.dd	TextNow	5

Summary | Messages | Call Logs | Contacts | Media Attachments

Account Contacts

Book Entries: 7

Communication References: 0

File References in Current Case

Path

/img_android_data.dd/data/com.viber.voip/databases/viber_data

/img_android_data.dd/data/com.viber.voip/databases/viber_messages

Other Occurrences

Accounts Browser - Contact Book

Browse | Visualize

Account	Device	Type	Items
1b397bcf-1455-4c0d-8736-29fe	android_data.dd	Device	201
+16784357227	android_data.dd	Viber	28
u8a7e68b3365c07e64fade456	android_data.dd	LINE	26
live:.cid.2d133a28af8d285f	android_data.dd	Skype	25
+16174534409	android_data.dd	TextNow	13
8:live:.cid.ab01b1b1545b16d9	android_data.dd	Skype	12
8:live:.cid.8f278e2e9be3ae57	android_data.dd	Skype	12
+17812314569	android_data.dd	Viber	11
100039669064643	android_data.dd	Facebook	11
2000107721184083	android_data.dd	IMO	10
ua9cf19436f475243938b1001e	android_data.dd	LINE	9
ub830497e95d7a2adf0610ff2a	android_data.dd	LINE	9
2000222005372359	android_data.dd	IMO	8
uaa130a5b1246ea6106f576cc	android_data.dd	LINE	6
live:btforensics1	android_data.dd	Skype	6
100000829961026	android_data.dd	Facebook	6
+16179441839	android_data.dd	TextNow	5
+15086235961	android_data.dd	TextNow	5
+15084359878	android_data.dd	Viber	5

5

Summary | Messages | Call Logs | Contacts | Media Attachments

Name	Email	Phone Number
Darlene Leigh	dleigh12@gmail.com	1 781-231-4569
Rob McIvan	mcivan33@yahoo.com	+1 508 435 9878
Shaina Brook		212-786-4321
Jake Smith	jake_smith@rcn.com	1 (412) 397 6868
Mom		+16178907272
Dad	acox56@verizon.net	+13024659870
Jack Spear		

Darlene Leigh

Properties

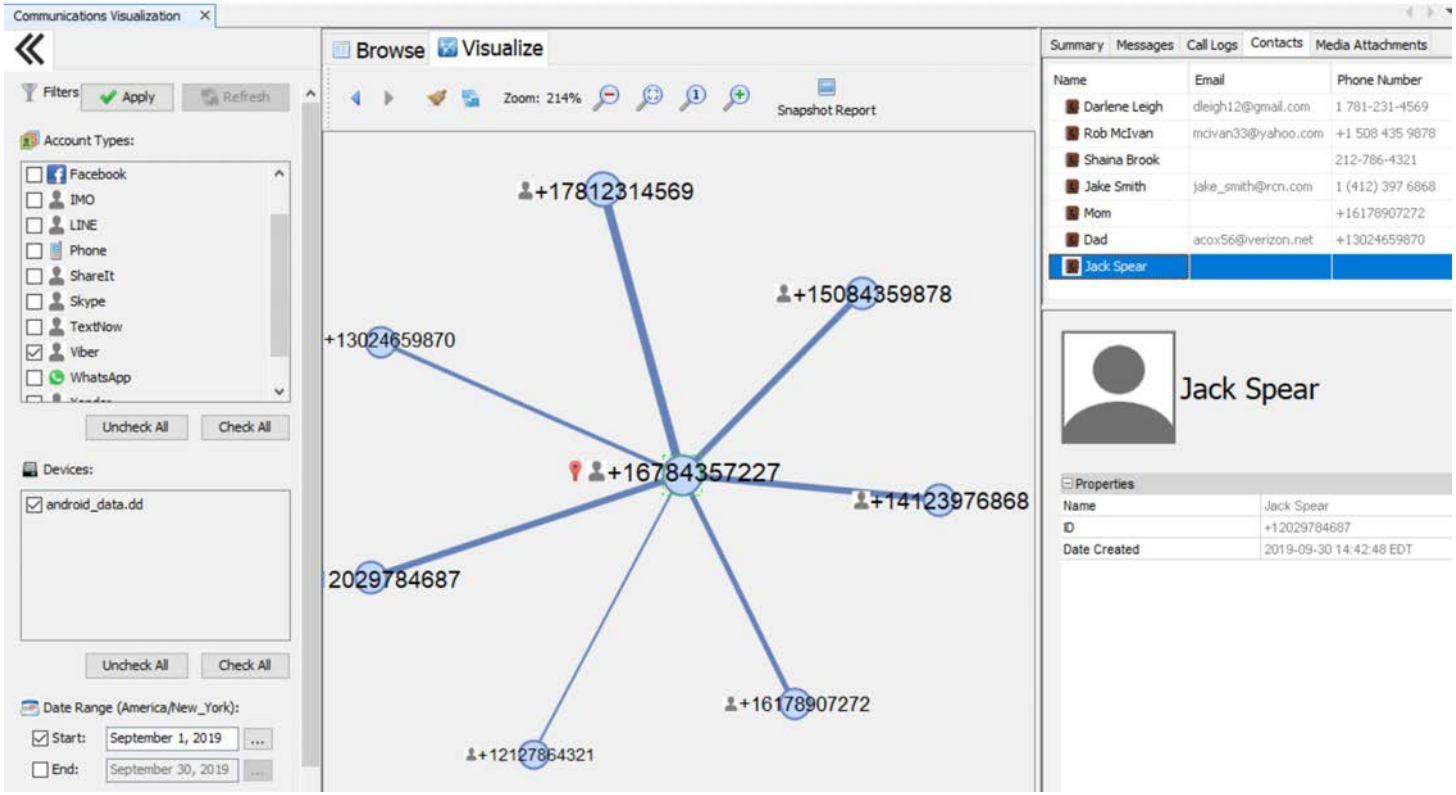
Name	Darlene Leigh
Phone Number	1 781-231-4569
Email	dleigh12@gmail.com
ID	+17812314569
Date Created	2019-01-12 14:47:47 EST

Communications Viewer - Visualizer

Graphical Visualization

- Helps identify more active accounts and clusters
- Link analysis

Communications Viewer - Walkthrough



Communications Viewer - Walkthrough

Browse | Visualize

Zoom: 214%

Snapshot Report

+17812314569

+13024659870

+16784357227

+12029784687

Summary | Messages | Call Logs | Contacts | Media Attachments

Showing Messages for Thread: I need money. A Lot. Come meet me at your fav bar 6 pm tomorrow.

Threads

Type	From	To	Date	Subject	Attachments
Messag	+12029784687	+16784357227	2019-09-30 14:...		0
Messag	+16784357227	+12029784687	2019-09-30 14:...		0
Messag	+12029784687	+16784357227	2019-09-30 14:...		0
Messag	+12029784687	+16784357227	2019-09-30 14:...		0

From: +12029784687 2019-09-30 14:48:50 EDT

To: +16784357227 Incoming

CC:

Subject:

Headers | Text | HTML | RTF | Attachments (0)

I need money. A Lot. Come meet me at your fav bar 6 pm tomorrow.

#OSDFCon
OPEN SOURCE DIGITAL FORENSICS CONFERENCE

How to Support New Apps

(Quick overview for developers)

The Need for Plugin Modules

- New apps are constantly being released and may not yet be officially supported.
- Apps change their database schemas and existing parsers may fail or not get all available data.

- You can help the community by writing and updating app parsers.

Why Build Modules in Autopsy

- Building a standalone parser requires:
  - Dealing with different inputs and finding the databases
  - Querying the databases tables
  - Storing, displaying, and reporting on the results.
- Building an Autopsy module allows you to focus on bullet #2.
  - It hides that the input is an image or file system collection
  - It provides UIs
  - It provides reporting
- All you need to think about is how to query a database

Expanding “Official Autopsy” Modules

- If you find that Autopsy’s support for an app needs to be updated, you can update its module.
- We’ve written them in Python to make it easy for the community to update.
- You can find the modules in the InternalPythonModules directory.
- Simply update the query and submit a GitHub Pull Request.

Expanding “Official Autopsy” Modules

This PC > OS (C:) > Program Files > Autopsy-4.13.0 > autopsy > InternalPythonModules > android

Name	Date modified	Type	Size
whatsapp	10/8/2019 1:10 PM	PY File	24 KB
skype	10/8/2019 1:10 PM	PY File	23 KB
line	10/8/2019 1:10 PM	PY File	22 KB
fbmessenger	10/8/2019 1:10 PM	PY File	18 KB
sbrowser	10/8/2019 1:10 PM	PY File	18 KB
textnow	10/8/2019 1:10 PM	PY File	17 KB
viber	10/8/2019 1:10 PM	PY File	17 KB
operabrowser	10/8/2019 1:10 PM	PY File	16 KB
imo	10/8/2019 1:10 PM	PY File	10 KB
contact	10/8/2019 1:10 PM	PY File	9 KB

Making Your Own Module

- If you want to support a new app, you can make your own module.
- It will be available to select in the list of Ingest Modules.
- To make a Python module, you need to:
  - Copy and paste our sample module.
  - Search for “TODO” and update things like the module name.
  - Write some code in the “process” method that will get called when the user picks your module.

- Go to “Writing Autopsy Python Module” talk for more details.

Building an App Parser (The old way)

1. Query the FileManager for specific database files. Repeat query for WAL/SHM files.
2. Save the database files to disk
3. Open the database
4. Query the database tables
5. Research which of the 40+ artifact types should be used (such as TSK_CONTACT)
6. Research which of the 100+ attribute types are relevant (such as TSK_PHONE_NUMBER)
7. For each entry:
    - Make an artifact with attributes
    - Make an “account” (for the Communications UI)
    - Make “relationships” between all of the accounts (for the Communications UI)
    - “Post” that the artifact was created, so that UI refreshes and it is indexed

Problem/Solution

Problems

- Many app modules have a lot of copy and pasted code and we want to make app modules as simple as possible.
- It is hard for writers to know which artifacts and attributes to use.
- It can be hard to get all of the account and relationship information correct.

Solution

- Build new classes to streamline the process and minimize code that modules need to have.

Building an App Parser (The new way)

1. Search for and open databases and associated WAL/SHM in a single method call.
2. Query the database tables
3. For each entry:
    a. Call a single method that creates artifacts, attributes, and relationships

No need to:

- Explicitly find and save WAL/SHM files
- Research all of the artifact / attribute types
- Learn about all of the communication-specific data types

New Classes

- AppSQLiteDB
  - Finds and opens application databases. Simplifies running queries
- CommunicationArtifactsHelper
  - Adds messages, call logs and other communication artifacts
- WebBrowserArtifactsHelper
  - Adds web cookies, bookmarks and other browser artifacts
- ArtifactsHelper
  - Adds GPS coordinates and other miscellaneous artifacts

No additional work is necessary to make data visible in the Communications UI or Results Tree.

Example: Finding Viber Databases

```
AppSQLiteDB.findAppDatabases(data_source, “viber_data”, True, “com.viber.voip”)
```

Database name

Folder name

This will return a list of databases with that name in the specified folder.

Example: Querying Viber Database

```
app_database.runQuery("SELECT phonebook.name,
                              phonebook.home_phone,
                              phonebook.work_phone,
                              phonebook.email
                       FROM   phonebook")
```

This will return a database cursor for the query results.

Example: Storing the Data


```
CommunicationHelper.addContact("John Doe", "413-362-1253", "",
        "512-126-2363", "john.doe@gmail.com")
```

Note: You may pass null or the empty string for data you don't know.

What's Coming Next?

- More app parsers
- Ability to import reports from other mobile forensics tools

- Better association between messages and attachments that are stored in some other part of the file system
- Adding more features to accounts
  - Linking accounts to a person
  - Mapping an account to all its known user names

Reach Out!

- If you have any development challenges, post a question on the forum:

  http://forum.sleuthkit.org

- If you have app requests, let us know now or on the survey.