9th Annual

#OSDFCon

OPEN SOURCE DIGITAL FORENSICS CONFERENCE

Messaging App Forensics with Autopsy

Brian Carrier

October 17, 2018 | Herndon, VA | Hosted by BASIS TECHNOLOGY

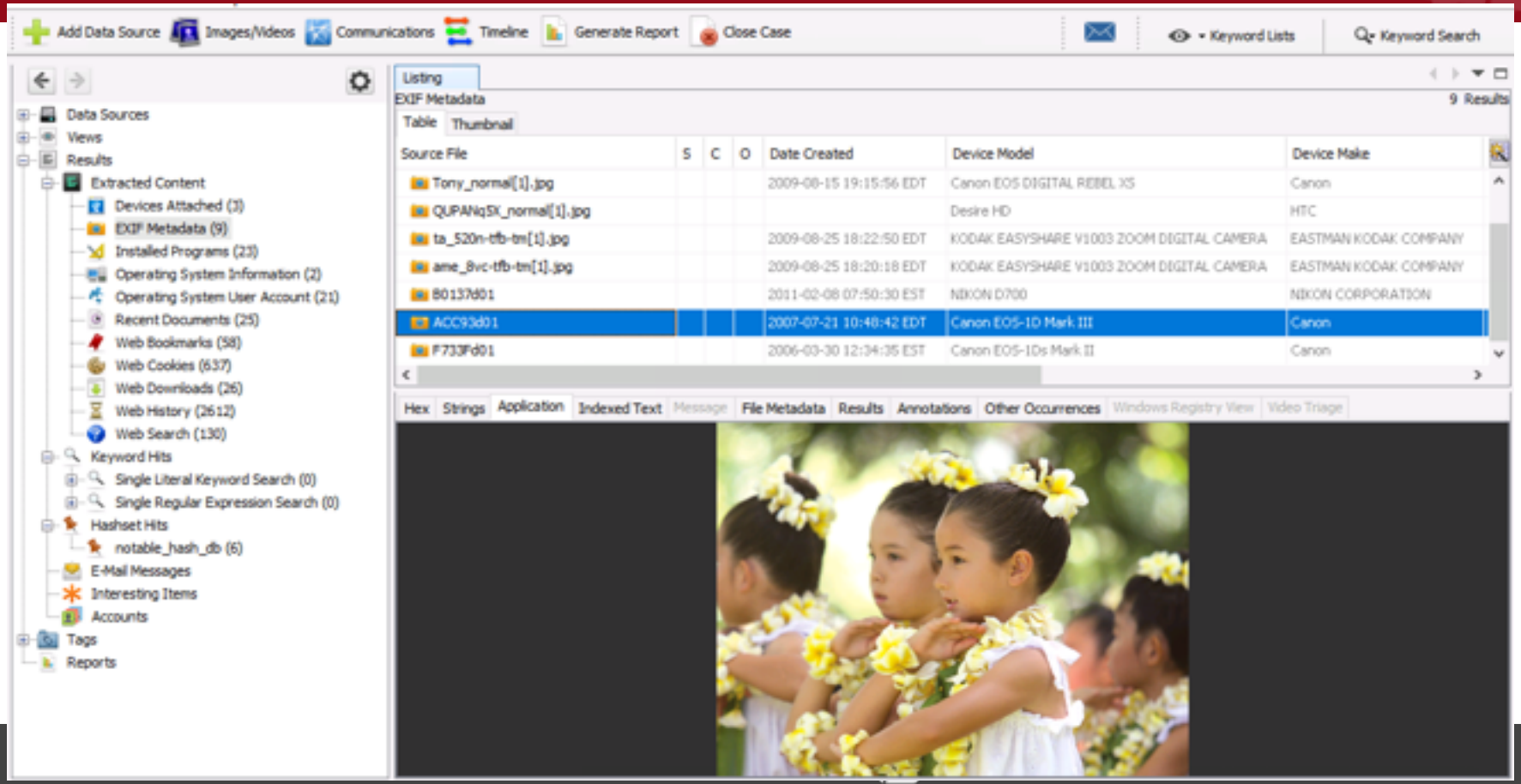
Motivation

Show users new features around messaging, email, and chats.

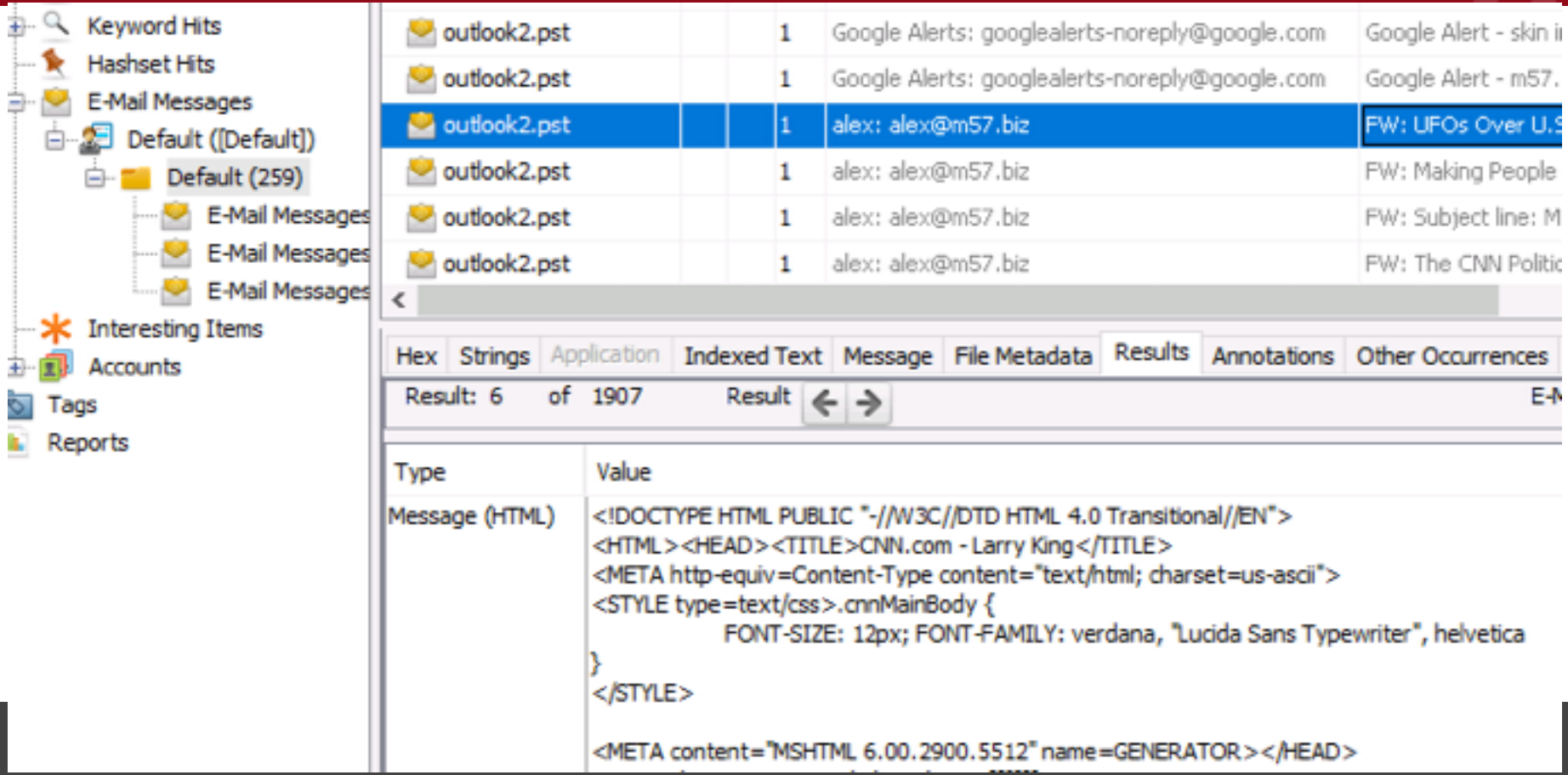
Show developers how to use the new infrastructure.

This work was all funded by DHS Science and Technology (S&T).

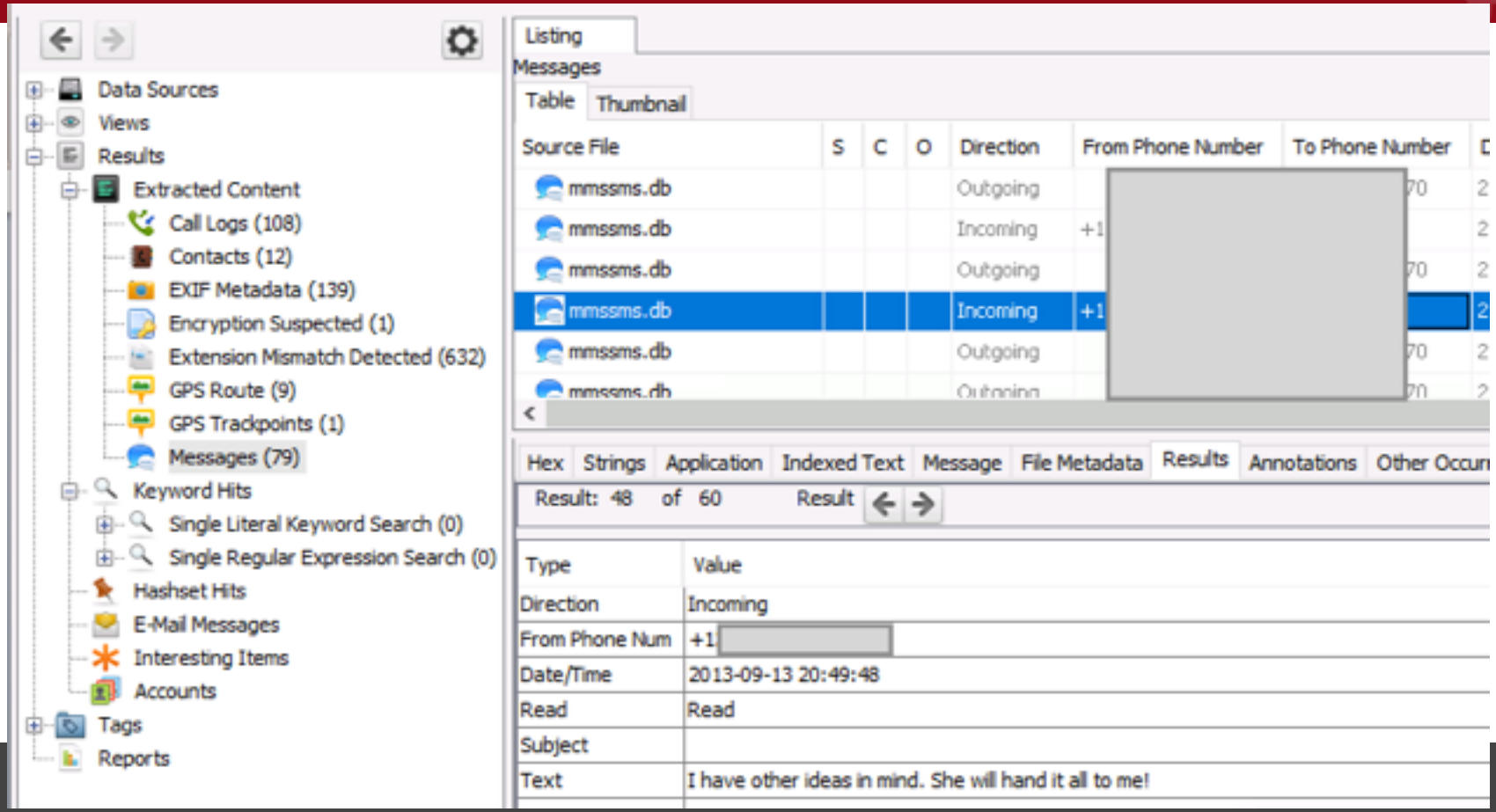
Main Autopsy UI



Email: The Old Way



SMS: The Old Way



Problem / Solution

Problem: Hard to sort through large amounts of messages

Why: Autopsy stores and displays things in generic ways so that it can support unknown data types. It's a general framework.

Solution: Store and display messages in more efficient ways.

What Questions Did We Focus On

Triage

- What are the most communicated with accounts?
- What kinds of media was this person sharing?
- What communication types was this person using?
- What were the most recent conversations?

Deep Dive:

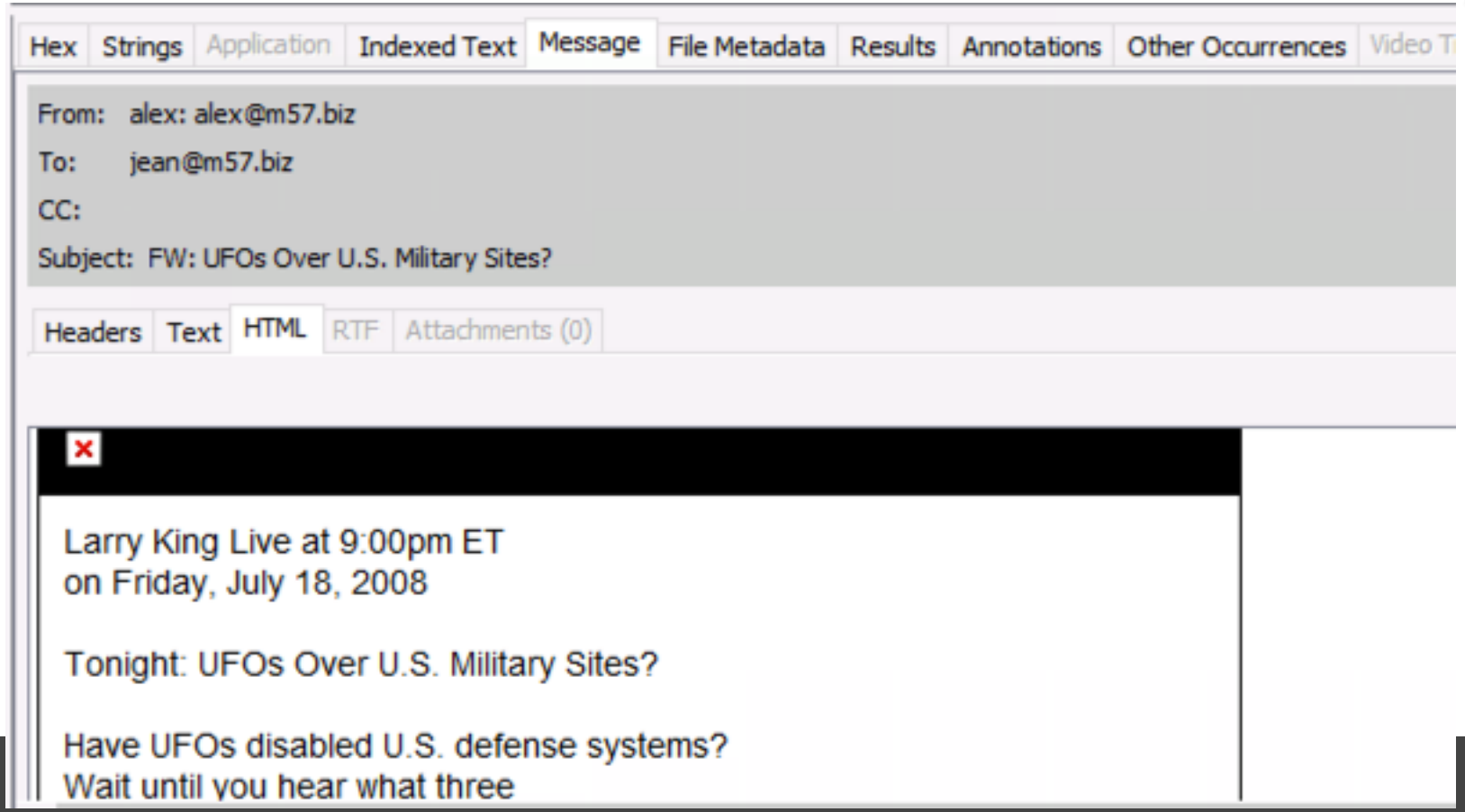
- What is the social network for this device?
- What are the themes and topics in the messages?

Change #1: New Content Viewer

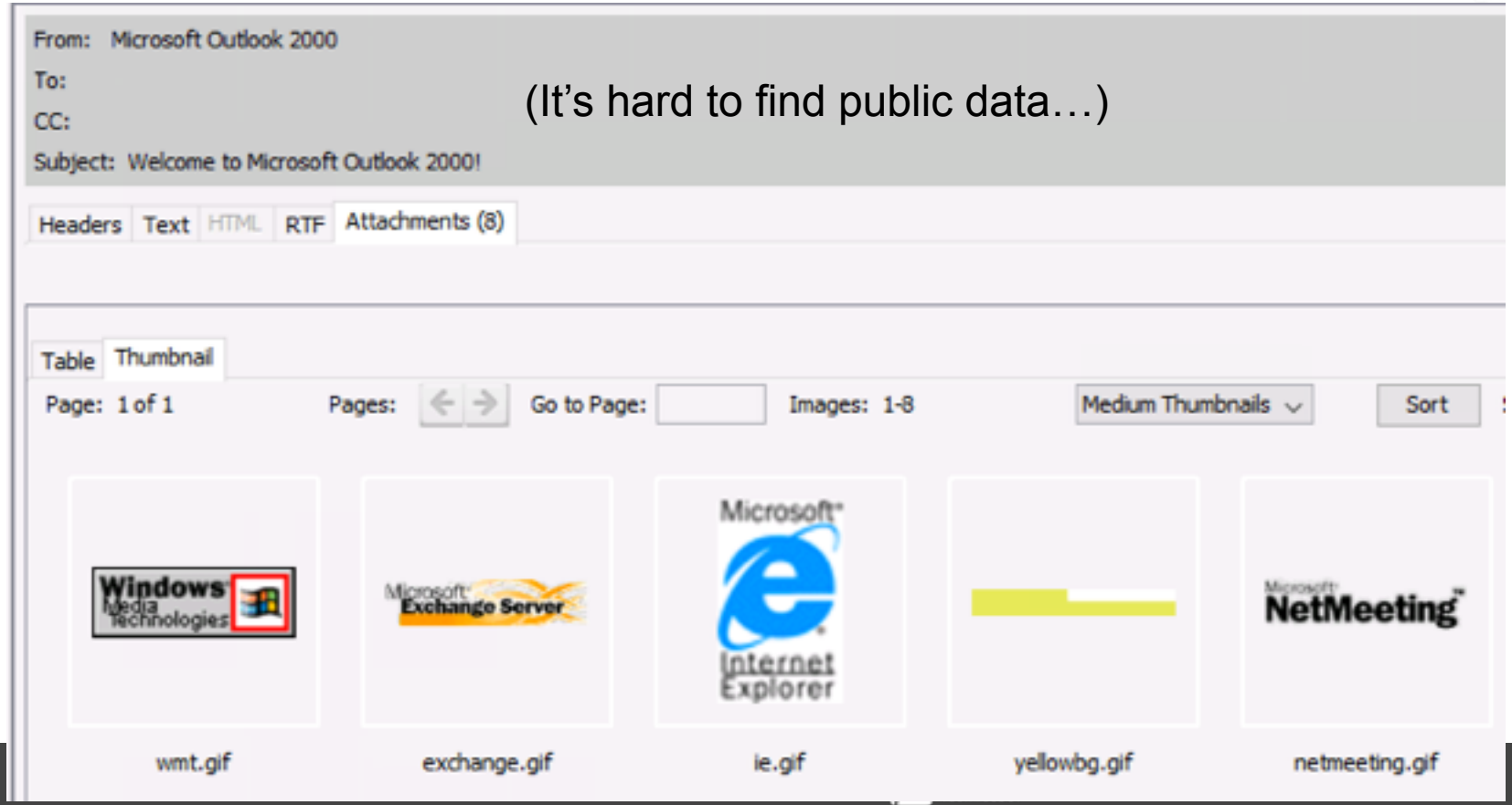
A new Message content viewer exists in the lower right. Shows:

- Headers
- Various formats of emails (HTML, RTF, etc.)
- Attachments
- Can load external images (if you opt-in)

New Content Viewer



New Content Viewer (Attachments)

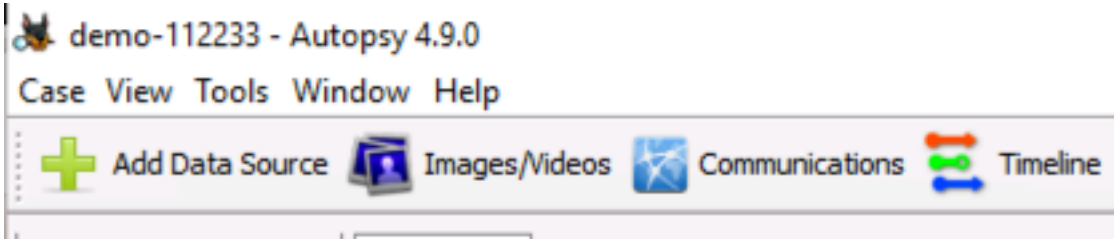


(It's hard to find public data…)

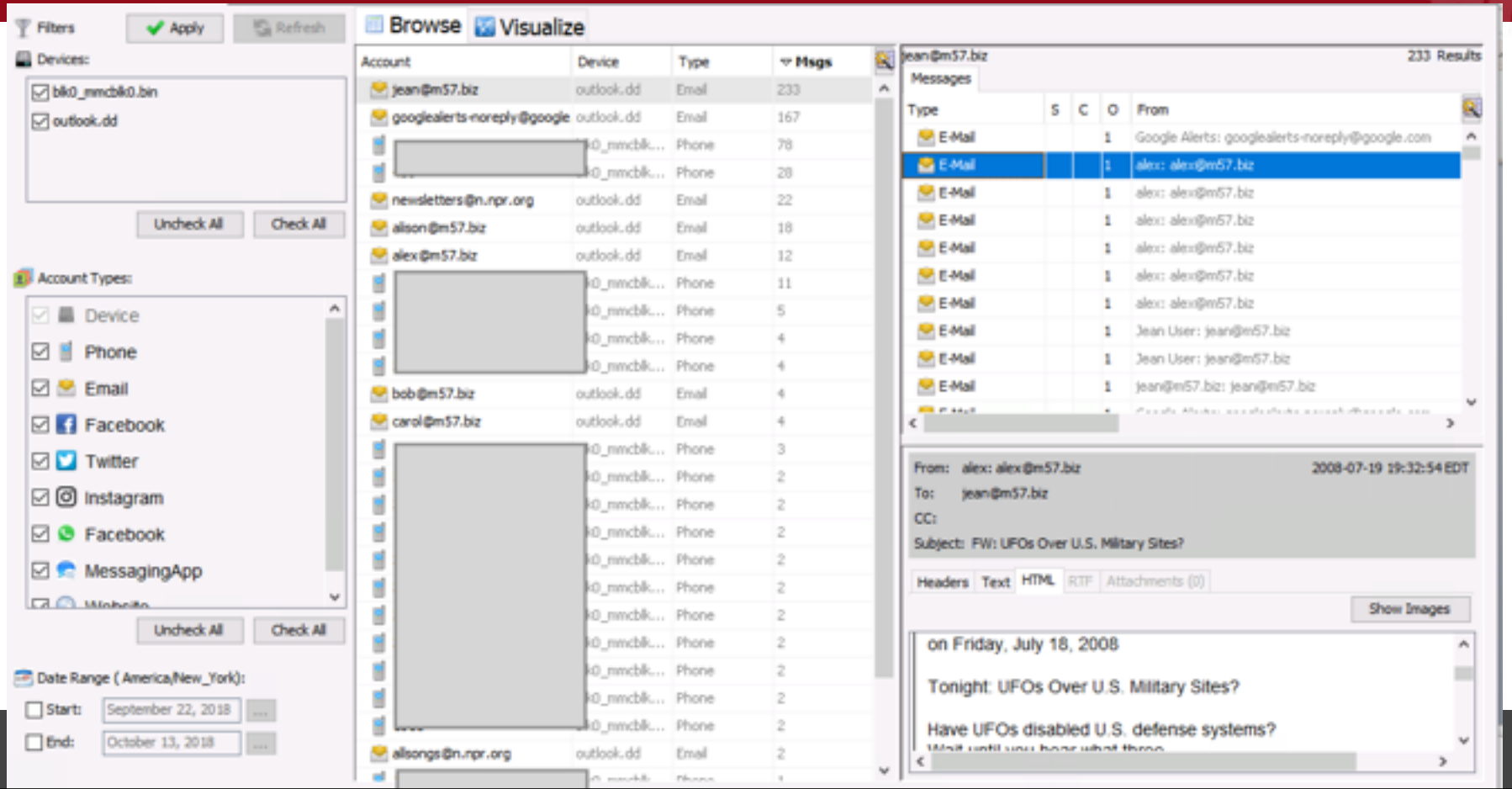
Change #2: New UI

The generic tree interface in Autopsy does not easily answer the previous questions.

So, we made a new one.



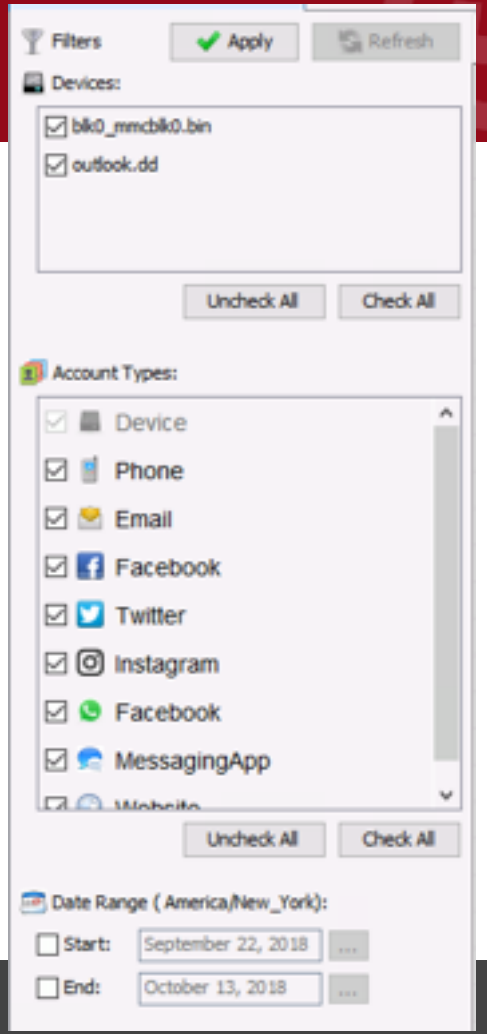
New Communications UI



UI Basics: Filters

Reduce the scope of messages and accounts based on:

- Dates
- Account Types
- Devices



UI Basics: View Accounts

Accounts that meeting filters are shown.

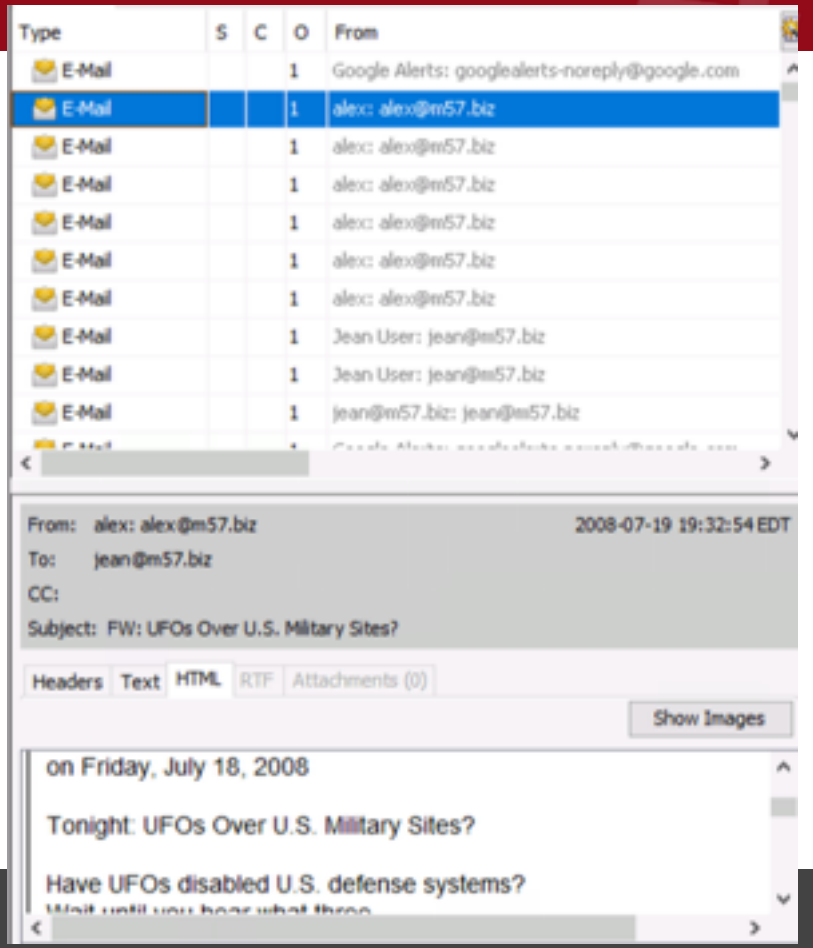
Sorted by number of relationships.

Only accounts with a relationship are shown (not random emails)

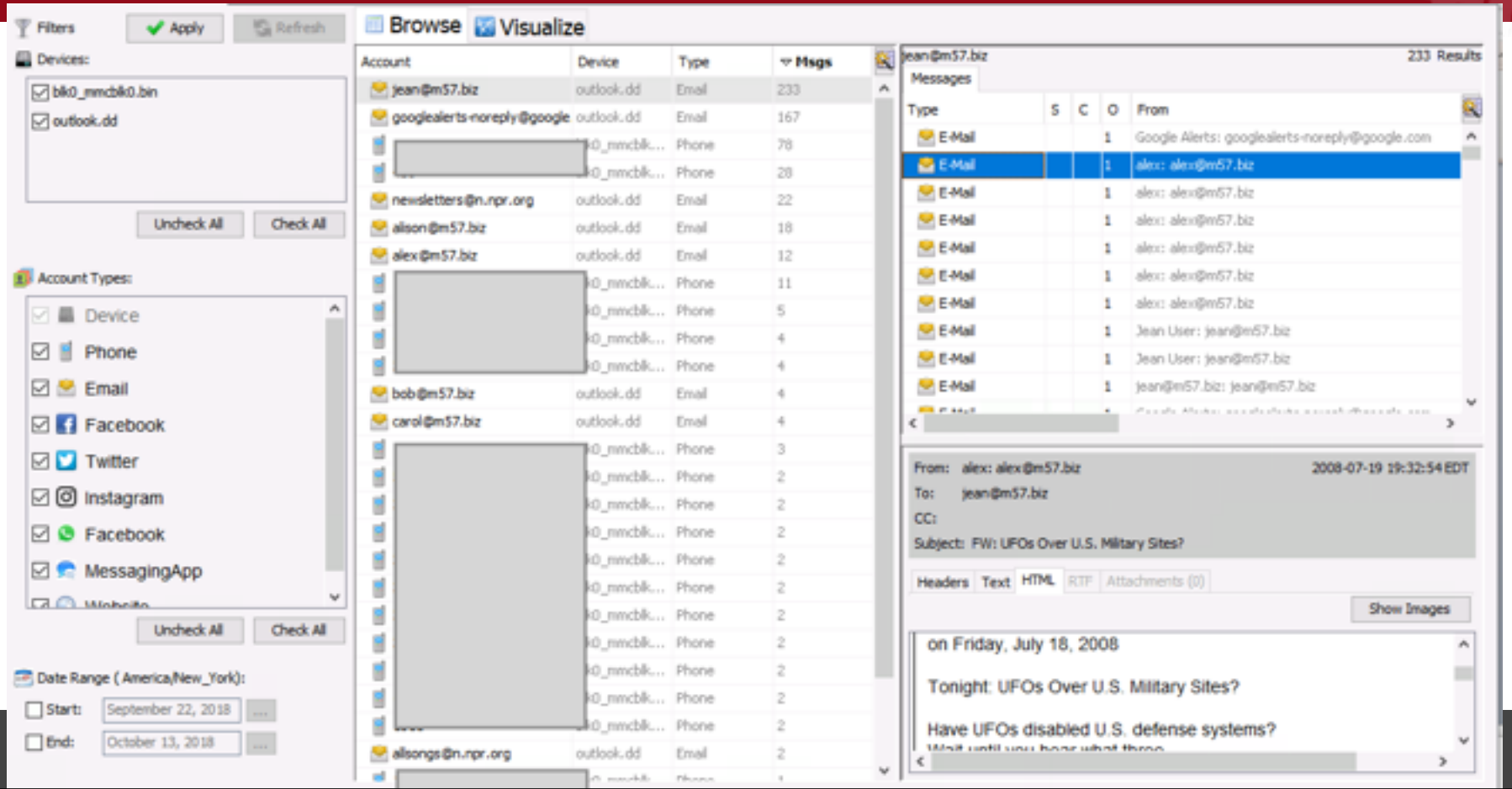
UI Basics: View Messages

Shows all messages to/ from an account.

Bottom shows message contents, attachments, etc.



New Communications UI



UI Basics: Link Analysis

Visualization of the data can be useful.

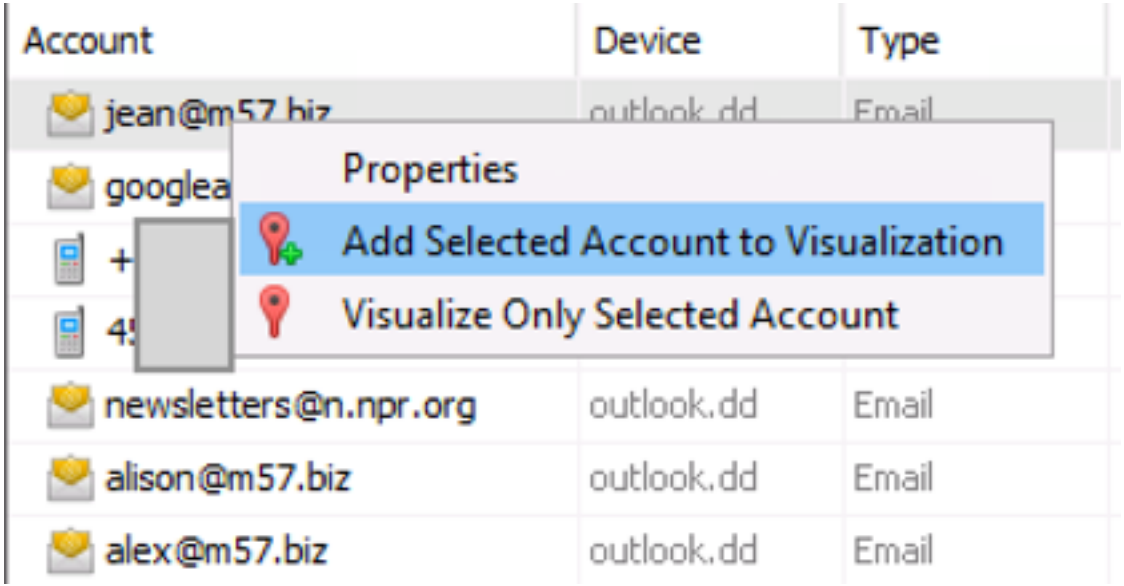
Link analysis diagrams can quickly become overwhelming.

We chose an “Opt-In” method:

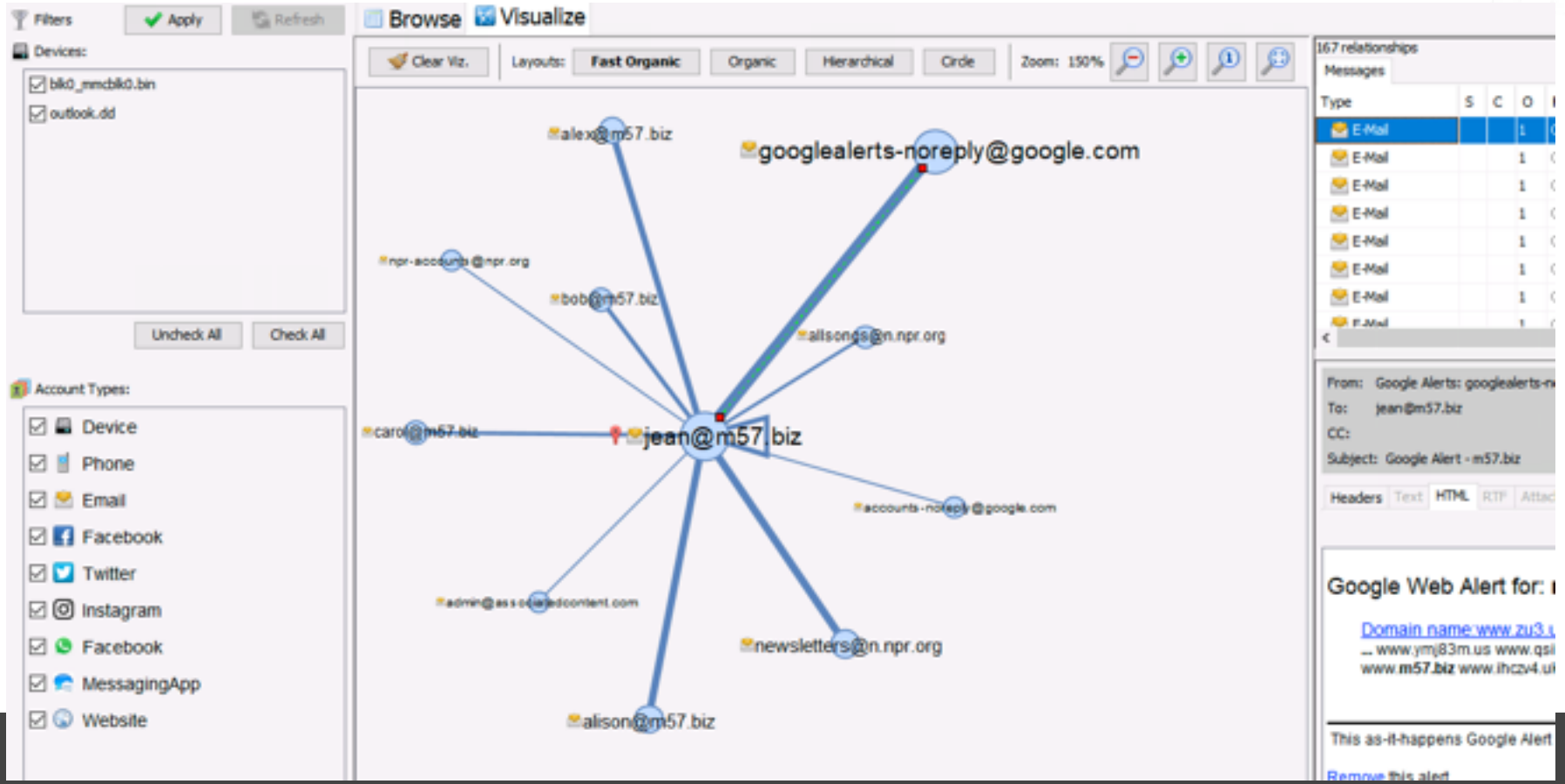
- Right click on an account and choose “Add Selected Account to Visualization”.
- That shows the account and all associated with it.

UI Basics: Link Analysis

Selecting an Account



UI Basics: Link Analysis

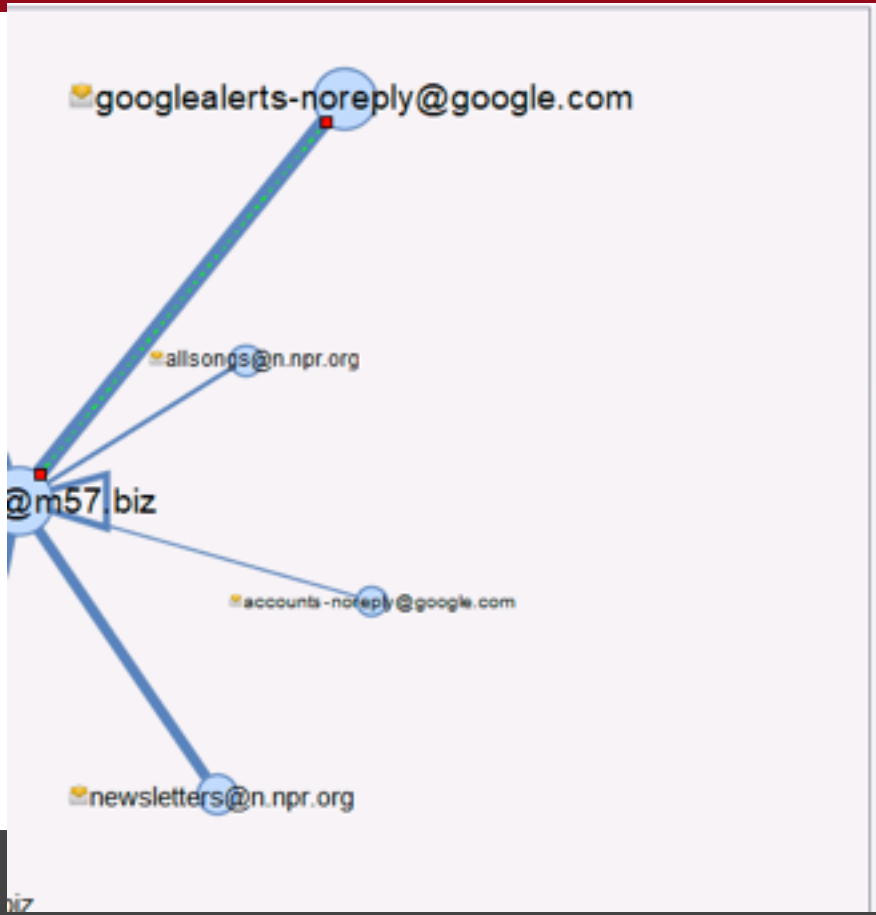


UI Basics: Link Analysis

Selecting a node shows all messages to and from that account.

Selecting an edge shows messages between the two accounts.

UI Basics: Link Analysis



Type	S	C	O	From
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal
E-Mail			1	Google Alerts: googleal

From: Google Alerts: googlealerts-noreply@google.com
To: jean@m57.biz
CC:
Subject: Google Alert - m57.biz

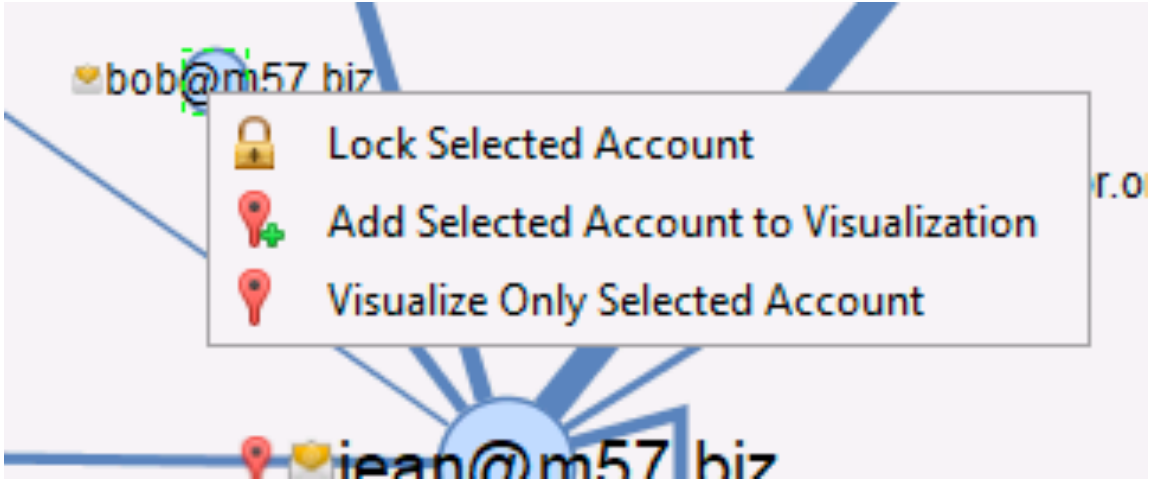
Headers | Text | HTML | RTF | Attachments (0)

Google Web Alert for: **m57.biz**

Domain name:www.zu3.uk The Constellati
... www.ymj83m.us www.qsif.br www.dtnlh.net w
www.**m57.biz** www.ihczv4.uk ...

Visualization: Adding More Accounts

Right click on an account to bring in all of its relationships:



Backend Framework

(Quick Overview For Developers)

Typical Scenario

You write a module to parse some kind of database to pull out messages.

You want to take advantage of this new UI.

Basic Approach

Create a Blackboard Artifact like you would for any other data type.

- Blackboard remains the primary place to store data from modules.

Add communications-specific data that points to the message artifacts.

Messages

Create a Blackboard Artifact

Example:

- Artifact Type: TSK_MESSAGE
- Attributes for recipients, content, headers, etc.

You can put whatever you have in here.

Accounts

Accounts have a type (such as email or Facebook) and an identifier (such as a@b.com)

You need to add an Account Instance for each file the account is found in.

```
senderAcct =
CommManager.createAccountFileInstance(EMAIL,
“a@b.com”, sourceFile);
```

NOTE: exact syntax is slightly more verbose

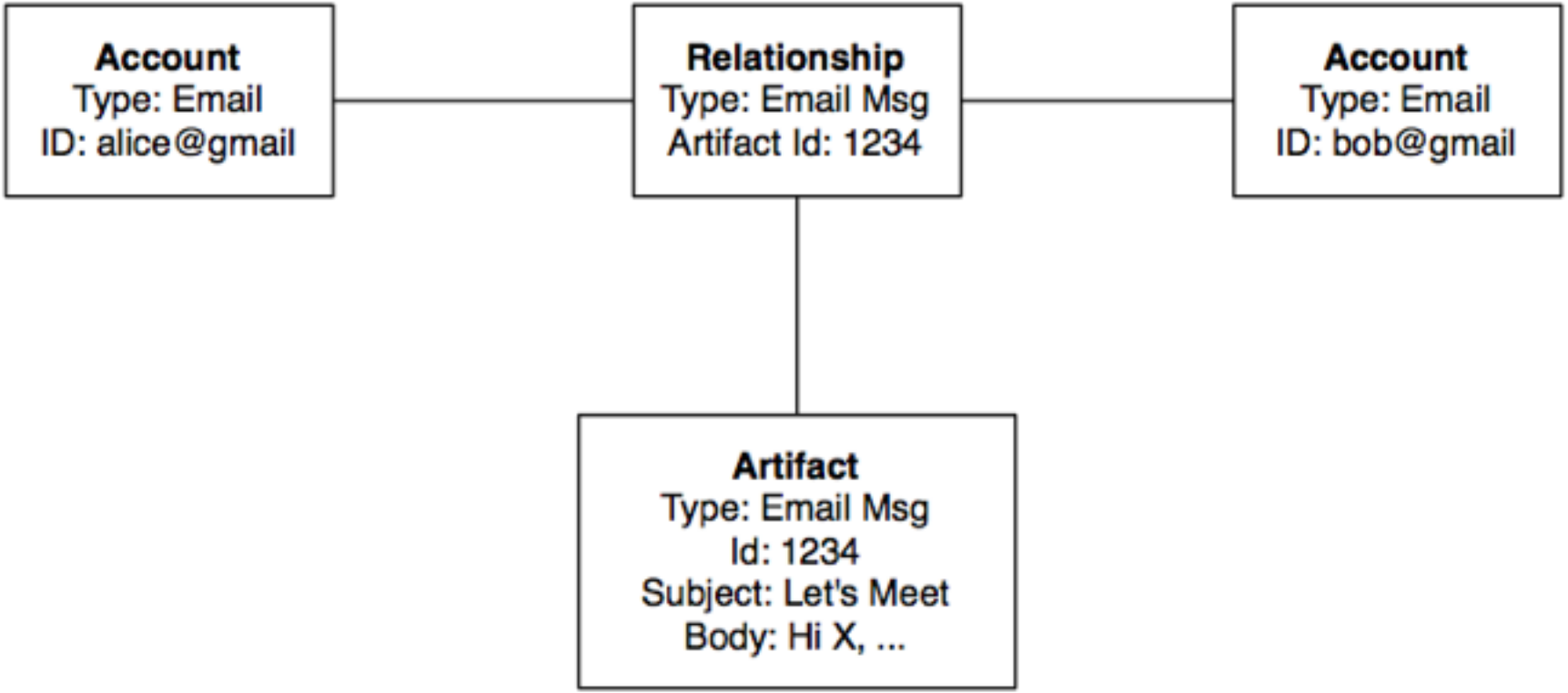
Relationships

A relationship occurs when two accounts “interacted” (i.e. sent message, made a call, in a contact book) at a given time.

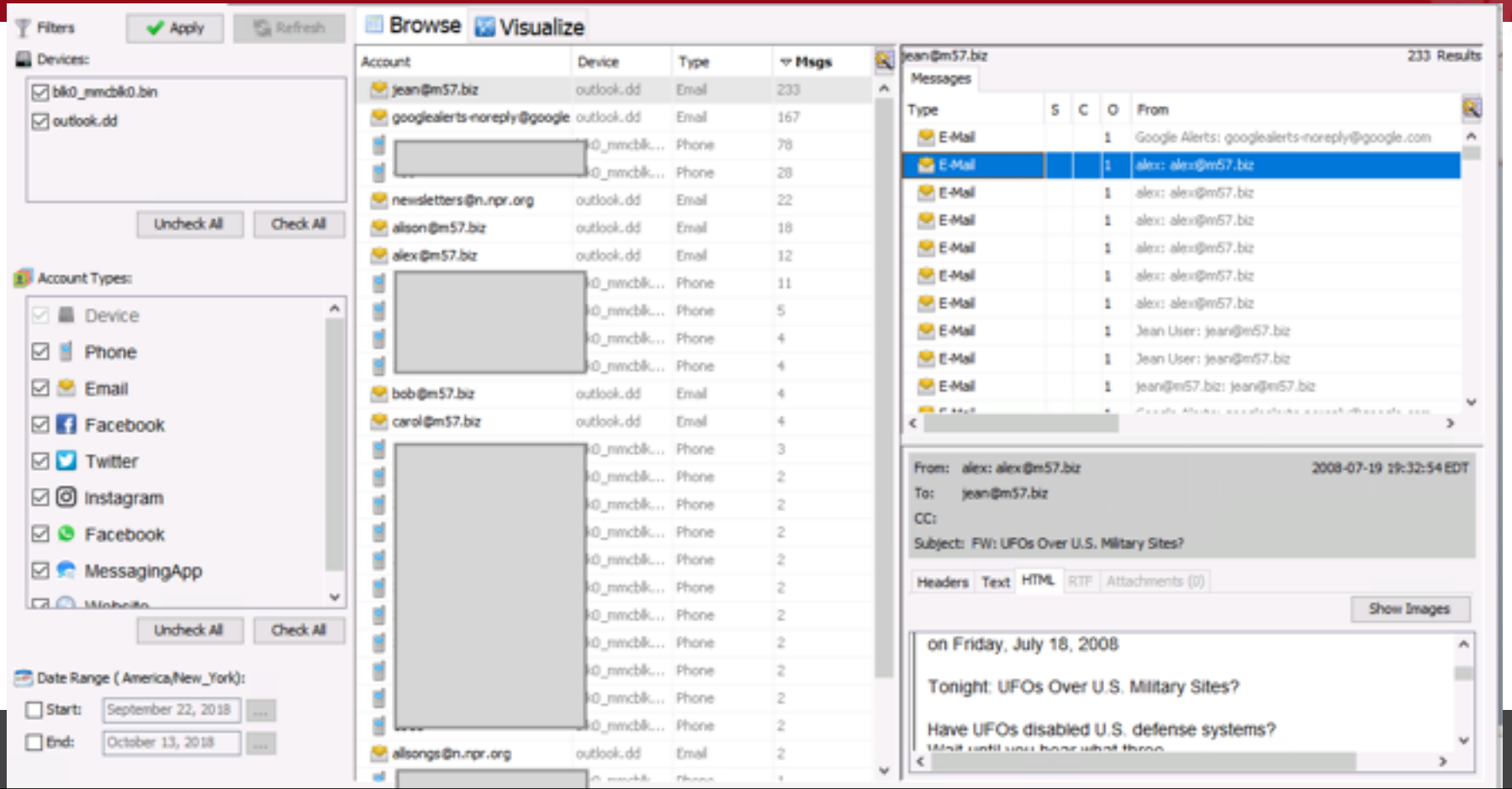
Add to the database with:

CommMgr.addRelationship(senderAcct, List of receiverAccts, messageArtifact, TYPE.MESSAGE, “2018-10-17…”)

Visualization



Viola: They Show Up



Call To Action!

We Need More Messaging Parsers

We built it, now we need more developers to come build parsers.

Build a Python parser for your favorite app.

Use the Android module or tutorials in Autopsy as a reference.

Next Steps

Roadmap

Scaling

Messages:

- Threading
- Thumbnails for attachments

Histogram for filtering

Automatically visualize the most communicated with accounts.

… LOOK AT

Questions?

brianc <at> basistech.com

Connect on LinkedIn

Provide future suggestions now,
during “Needs” session, or in survey.