



Tech Tuesday Workshop

So. Much. Data. How to Correctly Interpret Evidence from Smartphone Data

*Domenica Lee Crognale
& Heather Mahalik*

About Us

Heather Mahalik

- ▶ Senior Director of Digital Intelligence at Cellebrite
- ▶ SANS Senior Instructor and Course Author for FOR585
- ▶ SANS DFIR Curriculum Lead
- ▶ Involved with InfoSec/Forensics for 19 years

Lee Crognale

- ▶ Cybersecurity Engineer at ManTech
- ▶ SANS Certified instructor and Course Author for FOR585
- ▶ Involved with InfoSec/Forensics for 14+ years



Research and
validation is an
important skill in
every field

- MOBILE DEVICE FORENSIC ANALYSTS
- APPLICATION DEVELOPERS
- SECURITY RESEARCHERS

What we will cover

- ▶ Choosing the best test devices
- ▶ Methods for rooting and jailbreaking and why they matter
- ▶ Getting access to the data that tells the story
- ▶ SQLite databases (creating and querying)
- ▶ Creating test scenarios to answer important questions
- ▶ Making sense of test data
- ▶ The importance of validating your tools and findings
- ▶ Free tools/methods to get you started!

What you'll need to follow along:

- ▶ iBackupbot
 - ▶ Available for Windows or Mac OS
- ▶ AgentRansack
- ▶ Database Browser for Sqlite
 - ▶ Available for Windows or Mac OS
- ▶ iOS image file created for analysis
- ▶ A test device of your choice if you want to populate test data for testing after this session
- ▶ **All software and image file downloads have been made available on our TechTuesday dropbox link*
- ▶ **<https://for585.com/techtues>**

Testing and Validation

- ▶ You need to confirm how a certain artifact appeared in the file system
- ▶ Determine whether certain user interactions with the device record evidence of changes in the file system
- ▶ You are seeing conflicting information from the different tools you are using
- ▶ You want to validate your findings by creating real-time test scenarios

Choosing the right device



Can your device be Rooted or Jailbroken?

Modifying the OS in such a way that unofficial/unsigned code and applications can be installed and run

Android (root)

- ▶ *Also allows for elevated admin or root level(super user) privileges*
- ▶ Soft/shell temporary roots
- ▶ Full roots

iOS (jailbreak)

- ▶ Untethered
- ▶ Semi-untethered
- ▶ Semi-tethered
- ▶ Tethered

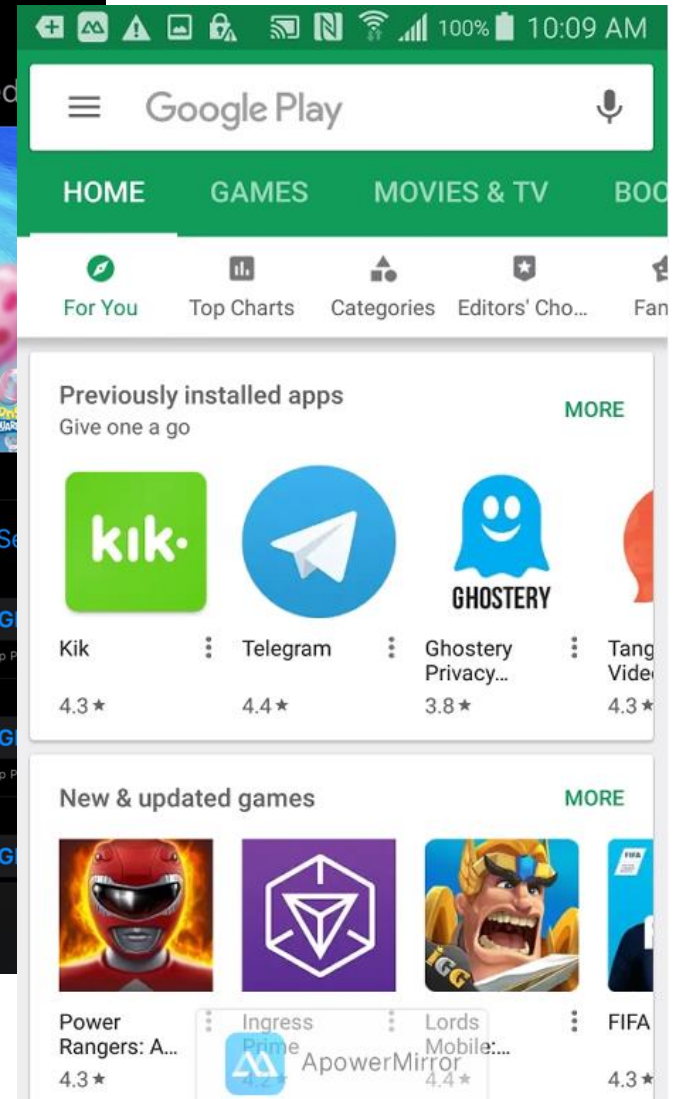
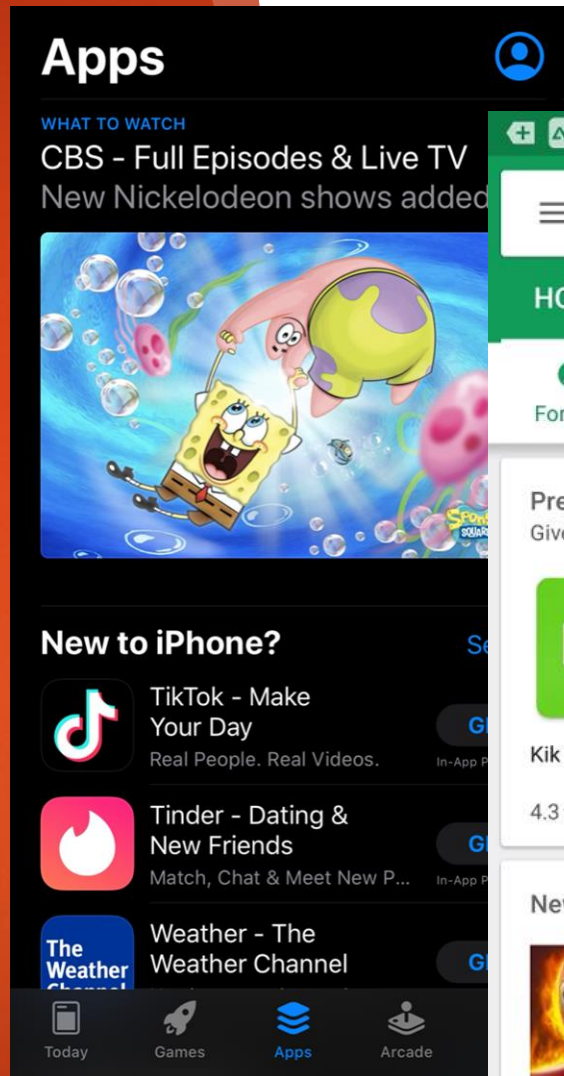
Android Considerations

- ▶ With root, your Acquisition tools will attempt to pull all of the physical partitions (which includes the entire **user data** partition)
 - ▶ Or you can do this using ADB command
- ▶ Nice to have a test device that has a PERMANENT root
- ▶ Newer devices and certain manufacturers/carriers are more restrictive
 - ▶ International models usually have less restrictions
- ▶ **Requires an unlocked bootloader**
 - ▶ Bootloaders can be locked by manufacturer or carrier
 - ▶ Purpose of the Bootloader: checks digital signature of original ROM so only approved Operating System is allowed to boot
 - ▶ Can be unlocked but will likely void warranties
- ▶ **Do your research before you buy/brick!**
 - ▶ You need exact matches for make/model/firmware and build number in most cases.
- ▶ Without root, you may be missing key files (SQLite databases or other files) that make up the bulk of the user-created data you're after

iOS Considerations

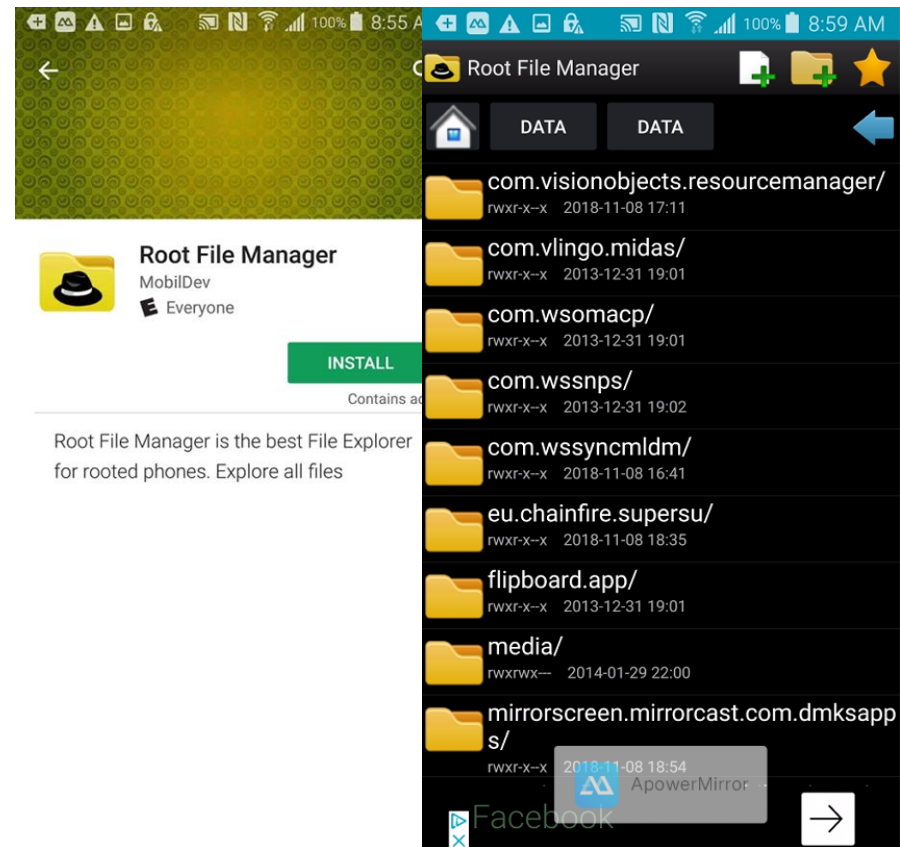
- ▶ Do you have a device that is vulnerable to checkm8?
 - ▶ Affects devices with A5 – A11 chips
 - ▶ May require that you update the firmware to the latest available version
 - ▶ Checkra1n jailbreak via SSH or Cellebrite's checkm8 dump
 - ▶ Support for iPhone models 5s through X
- ▶ New hardware is not supported so we're back to jailbreaks that exploit firmware vulnerabilities
 - ▶ <https://theiphonewiki.com/wiki/Jailbreak>
- ▶ iOS is faster at phasing out hardware/software combos
 - ▶ Don't expect these vulnerable checkm8 devices to be around forever ☹️
- ▶ Applications require (a very current) minimum firmware version for installation

Accessing the data of interest



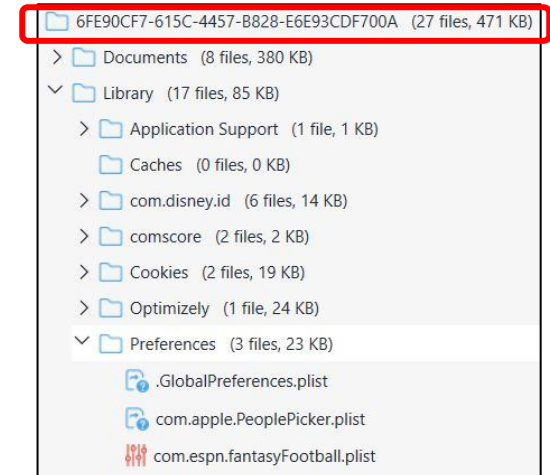
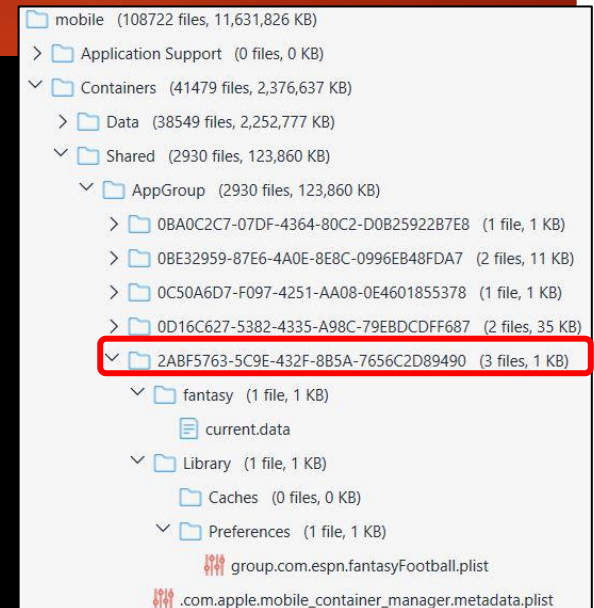
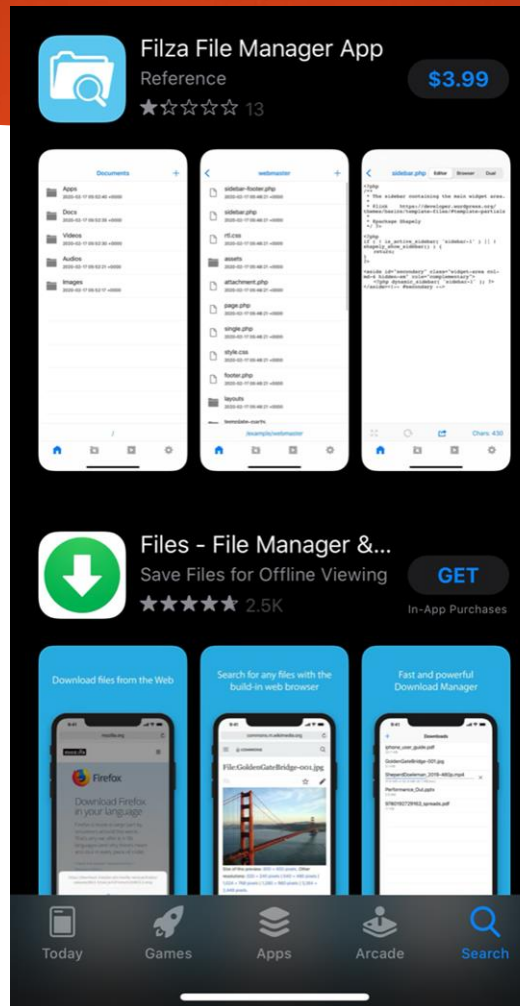
Android options:

- ▶ Install a File manager application to explore the file system and pull out files of interest.
 - ▶ Root File Manager (by MobilDev) is one of my favorites
 - ▶ Pull the entire application folder and data from other common storage locations
 - ▶ USERDATA/data/<application>
 - ▶ Physical or emulated SD cards
 - ▶ USERDATA/media
- ▶ Copy data out to a location that is accessible
- ▶ ADB commands can be used to pull application specific folders from the device (requires USB debugging)



iOS options:

- ▶ File Explorer tools are also available for iOS
 - ▶ Pick one that normalizes the app directory name (ie. Filza)
- ▶ Copy out application and shared app group directories
 - ▶ Paste them in a location that is accessible via USB
- ▶ Filesystem can be accessed from Mac terminal utilizing SSH (root/alpine)



Digging into
the
application
directory for
files of interest



SQLite Databases

- ▶ Most common format for storage of user-created items on smartphones
- ▶ Platform agnostic, supported by many scripting languages
- ▶ Databases consist of Tables, Rows and Records
- ▶ Input fields are assigned a recommended affinity type
 - ▶ NULL, integer, real, text, BLOB
- ▶ Primary Keys are used to illustrate uniqueness of records and will aid in joining multiple tables together
- ▶ Deleted data (pages) are often recoverable

ZKIKATTACHMENT (184)	ZUSER ▼	ZRECEIVEDTIMESTAMP ▼	ZTIMESTAMP ▼	ZBODY
ZKIKATTACHMENTEXTRA (178)	37	497800782.987472	497800782.474	Welcome to Kik, the super fast smartphone messenger!!
ZKIKATTRIBUTE (3)	41	497803811.6537	497803811.6537	questions, let me know. I'll do my best ☺
ZKIKCHAT (5)	41	497804389.586069	497804196.154	You started chatting with Ace
ZKIKCHATEXTRA (5)	41	497805068.863391	497805067.896	Hey lloyd, so glad we're finally in touch
ZKIKMESSAGE (558)	41	497805068.963701	497805067.915	7cbf883b-8672-44e0-97fe-c3705e75f7c7
ZKIKMESSAGEEXTRA (4062)	41	497805118.132724	497805118.132724	WHAT do you think of this☺ picture?
ZKIKPUBLICGROUPS (1)	41	497805427.726313	497805427.726313	I just sent you one of my current laptop
ZKIKUSER (292)	41	497805442.399056	497805442.399056	Hello microphone
ZKIKUSEREXTRA (292)	41	497812324.589263	497812324.156	Test chat from ace to lloyd
Z_4MESSAGES (558)	41	497813709.744746	497813693.037	Leaved a kik picture too
Z_9ADMININVERSE (5)				
Z_9BANSINVERSE (0)				

Hands-on
EXERCISE:

Let's build a
Database



SQLite Queries

- ▶ Queries begin with SELECT and always select at least 1 COLUMN from at least 1 TABLE
- ▶ Pay attention to comma placement/spelling
- ▶ Recognize and Convert timestamps to common formats
- ▶ **CAVEAT:** This is not a forensic tool so you will not see deleted records

SELECT

ZUSER,

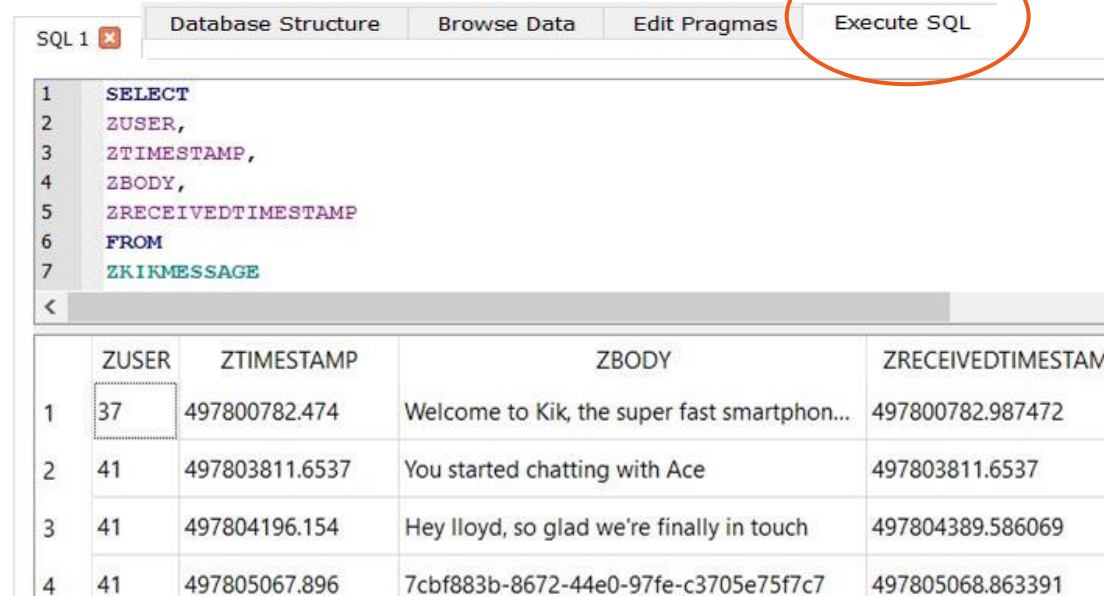
ZTIMESTAMP,

ZBODY,

ZRECEIVEDTIMESTAMP

FROM

ZKIKMESSAGE



SQL 1

Database Structure Browse Data Edit Pragas **Execute SQL**

```
1 SELECT
2 ZUSER,
3 ZTIMESTAMP,
4 ZBODY,
5 ZRECEIVEDTIMESTAMP
6 FROM
7 ZKIKMESSAGE
```

	ZUSER	ZTIMESTAMP	ZBODY	ZRECEIVEDTIMESTAMP
1	37	497800782.474	Welcome to Kik, the super fast smartphon...	497800782.987472
2	41	497803811.6537	You started chatting with Ace	497803811.6537
3	41	497804196.154	Hey lloyd, so glad we're finally in touch	497804389.586069
4	41	497805067.896	7cbf883b-8672-44e0-97fe-c3705e75f7c7	497805068.863391

SQLite: Timestamp Conversions

UNIXEPOCH

- The number of seconds since 01/01/1970 00:00:00
- **datetime(time COLUMN name, 'unixepoch')**

UNIXEPOCH in milliseconds

- The number of milliseconds since 01/01/1970 00:00:00
- **datetime(time COLUMN name/1000, 'unixepoch')**

Mac Absolute

- The number of seconds since 01/01/2001 00:00:00
- **datetime(time COLUMN + 978307200, 'unixepoch')**

Chrome

- The number of microseconds since 01/01/1601 00:00:00

SQLite: Timestamp Conversions

datetime (ZTIMESTAMP + 978307200, 'unixepoch'),

```
1 SELECT
2   ZUSER AS "User",
3   datetime(ZTIMESTAMP+ 978307200, 'unixepoch', 'localtime') AS "Timestamp",
4   ZBODY AS "Message",
5   datetime(ZRECEIVEDTIMESTAMP+ 978307200, 'unixepoch', 'localtime') AS "Message Received"
6 FROM ZKINMESSAGE
```

Timestamp
497800782.474
497803811.6537
497804196.154
497805067.896
497805067.915

	User	Timestamp	Message	Message Received
1	37	2016-10-10 09:59:42	Welcome to Kik, the super fast smartphone messenger! ...	2016-10-10 09:59:42
2	41	2016-10-10 10:50:11	You started chatting with Ace	2016-10-10 10:50:11
3	41	2016-10-10 10:56:36	Hey lloyd, so glad we're finally in touch	2016-10-10 10:59:49
4	41	2016-10-10 11:11:07	7cbf883b-8672-44e0-97fe-c3705e75f7c7	2016-10-10 11:11:08
5	41	2016-10-10 11:11:07	WHAT do you think of this picture?	2016-10-10 11:11:08

Hands-on
EXERCISE:

Querying the
database to
make sense of the
data



QUERY EXERCISE

- ▶ Locate the **sample_database.db** from <https://for585.com/techtues>
- ▶ Generate a query that returns the following:
 - ▶ First Name
 - ▶ Last Name
 - ▶ Preference
 - ▶ Date received
- ▶ Answer some questions about the dataset by using different filtering techniques

Best Practices for Approaching Application Testing/Validation

- ▶ Install the application of interest
- ▶ Start generating user data (make calls, send messages, etc.)
- ▶ Find and review the database that contains the information you created *Agent Ransack which will be discussed later can be useful for locating the database(s)
- ▶ Not all of your columns may be populated. Research the application on Apple/Google's store and make sure you utilize all of the available functionality of that app
- ▶ Create more test data and re-examine your database!

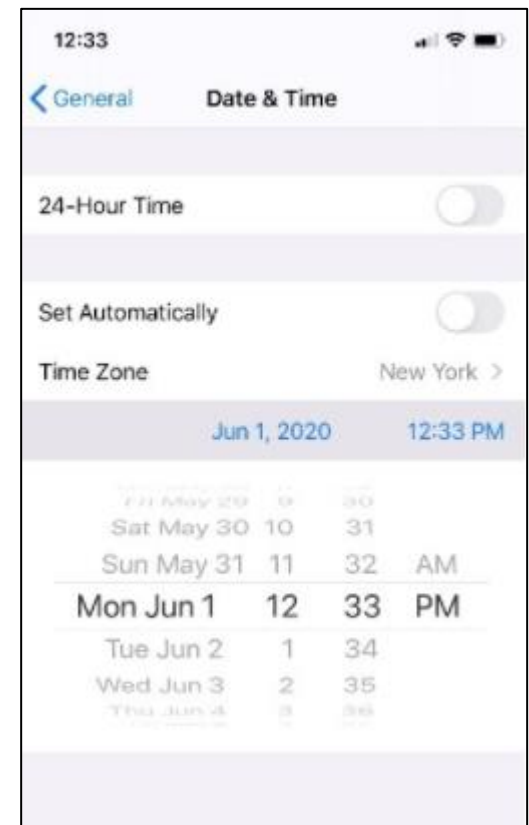
Considerations for Testing

- ▶ Has it been researched already?
 - ▶ Blogs, DFIR Review, etc.
 - ▶ Can you simply validate it?
- ▶ What are you trying to answer?
- ▶ Don't be afraid to ask for help
- ▶ Don't be afraid to do it yourself!
- ▶ Control your test environment
 - ▶ Create simple data – easy for you to analyze
 - ▶ Ensure you can extract the data
- ▶ Have someone in DFIR review it for you
- ▶ Share your findings!
 - ▶ No public blog, no problem – ask us to host it!



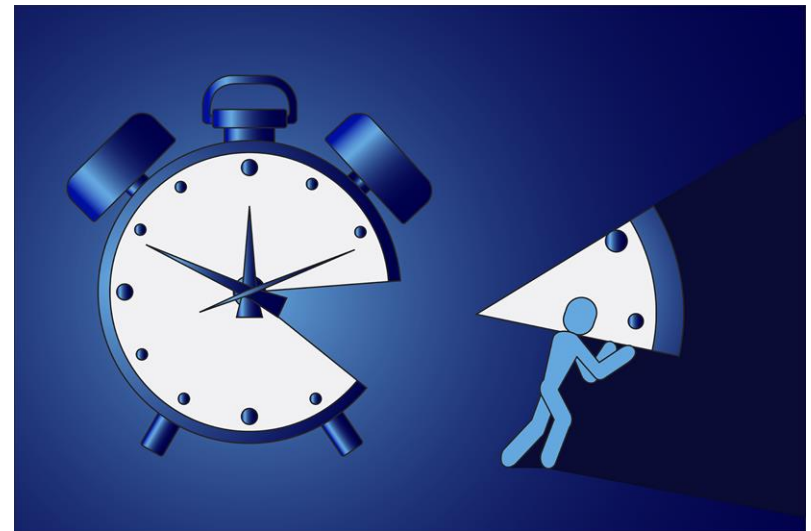
Scenario: Timestamp Manipulation

- ▶ It appears that data on the device does *not* fit the claims of when the user was on their device.
- ▶ How did this happen?
- ▶ Considerations:
 - ▶ What if someone changes the clock on the iPhone?
 - ▶ If they make a call, what will the timestamp look like?
 - ▶ If they send a text, what will the timestamp look like?
 - ▶ What if they change it back before it lands on your desk?



Hands-on
EXERCISE:

Let's create
test data



Part 1: Creating Test Data

NOTE: If you do not have an iOS device, do not worry – a dataset will be provided for you to use in the real lab.

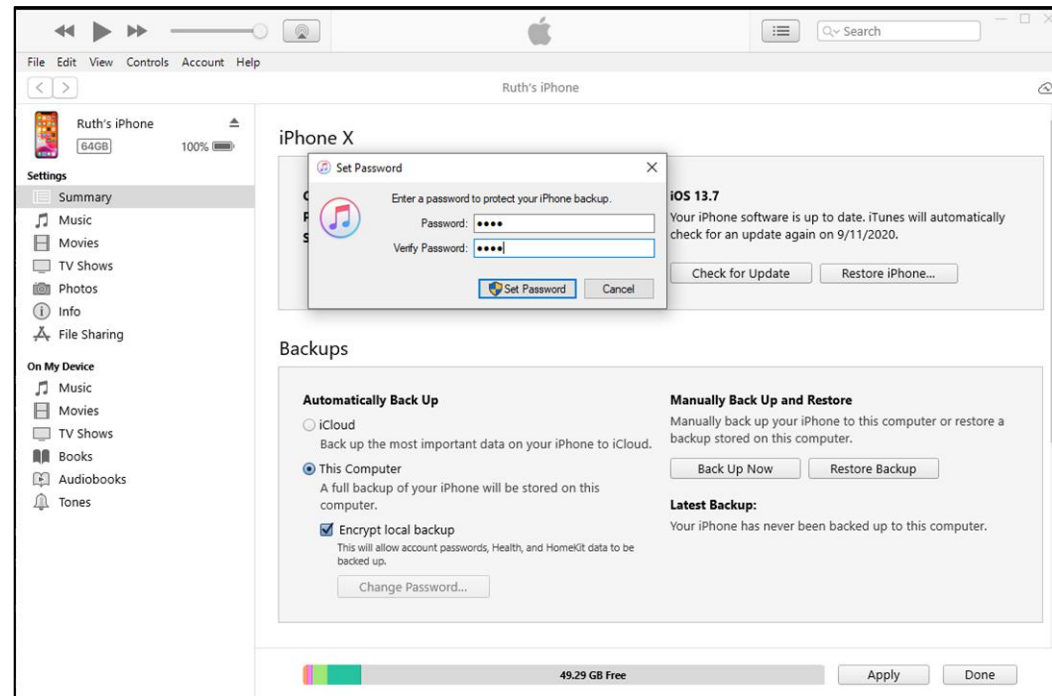
- ▶ Note the time on the iOS device
- ▶ Go to Settings>General>Date&Time and change the clock to Sept 1, 2020 at 12:00 PM New York timezone.
 - ▶ Make sure **Set Automatically** is switched off or this will not work!
- ▶ Dial a number you know and record the timestamp
caveat - *a commercial tool may be required*
- ▶ Send a text to someone you know and type “**testing time change on iOS**” and note the timestamp
- ▶ When noting timestamps – we recommend noting actual time and date as well as the time and date that you altered the phone to reflect! (i.e 9/1/2020 at X (altered time) & 9/8/20 at X (actual time))

Extracting the Data

****Try this after this session!**

- ▶ Launch iTunes
- ▶ Connect the device
- ▶ Pay attention to the iPhone for password prompts – you need to do this to trust and to encrypt the backup
- ▶ Select **This Computer** and **Back Up Now**

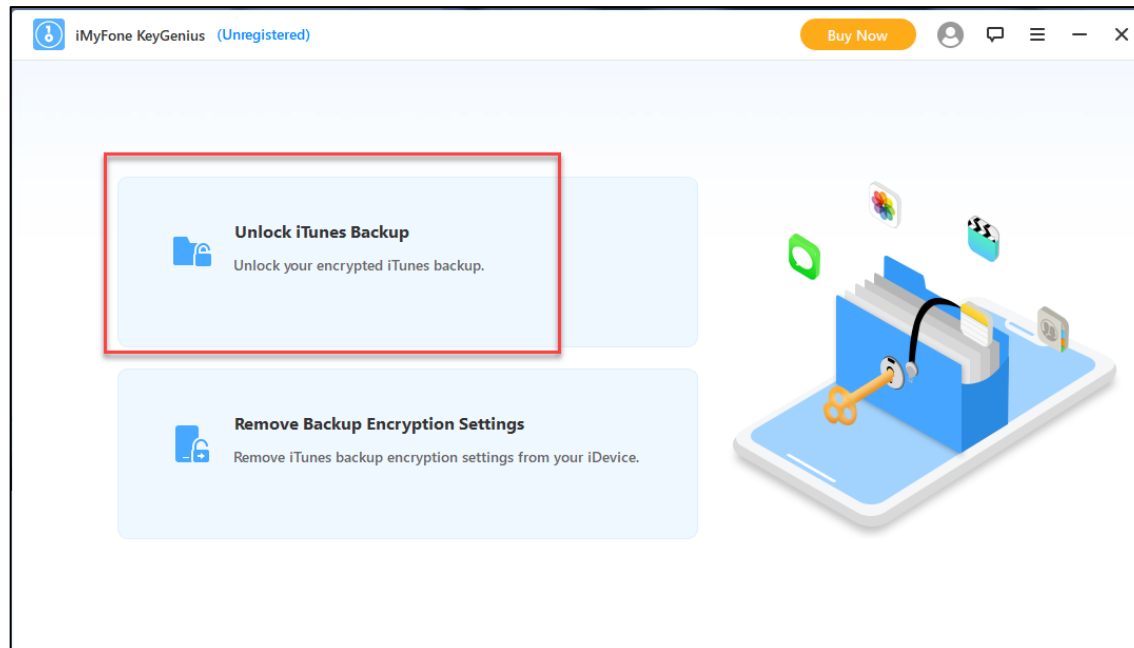
Caveat - for iOS 13 and later you **MUST** encrypt the local backup in order to extract Calls, Health, Safari History, Maps and Wallet.



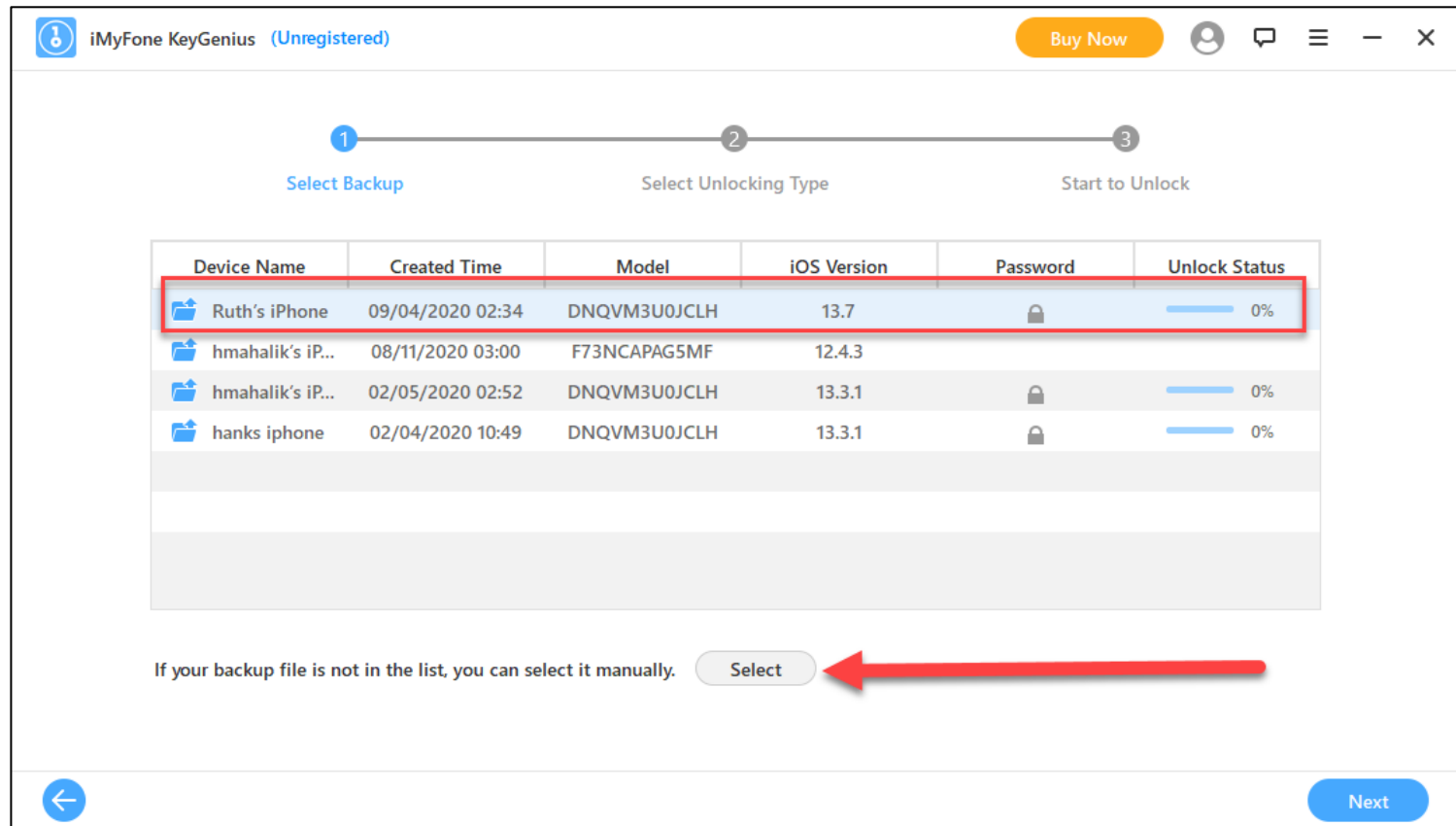
Decrypting the Backup (1)

**This can be done by almost all commercial tools

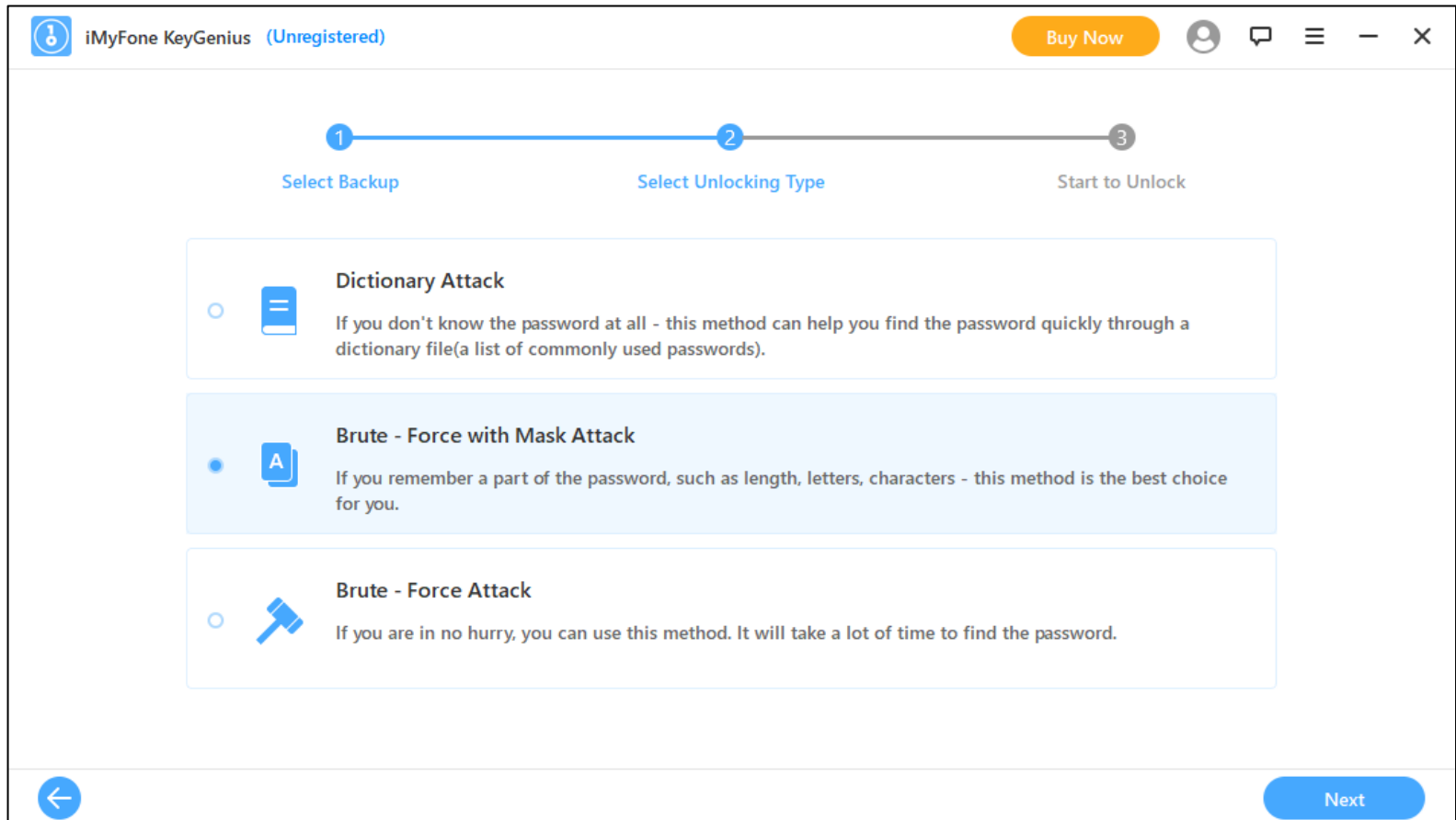
- ▶ Free options exist, but may be hard to find
- ▶ This one is about \$50



Decrypting the Backup (2)



Decrypting the Backup (3)



Decrypting/Parsing the Extracted Data

- Commercial tools make it easy, but they are not free

Apple iOS iTunes (Back...)

Learn more

Want to take advantage of key capabilities to advance your case?
Learn with these Tips, Tricks and Tutorials

A closer look at iOS time

iTunes backup encryption password

The extraction is encrypted. To continue with the decoding process, enter the iTunes backup encryption password and click OK.

If you do not know the iTunes backup encryption password, click "Load from file" to load passwords from a text file (dictionary). The file must include a list of passwords, with each password on a separate line. This process runs locally on your computer, and may take some time to complete.

Notes:

1. The iTunes backup encryption password is required here to access encrypted backups, and is different from the iPhone device PIN code.
2. If the iTunes encryption password is not available, approach Cellebrite CAIS for a possible encryption bypass solution.

Load from file OK Cancel

Take a moment to learn more

10 Common Mistakes Many Examiners Make

This episode of Carved from Unallocated covers 10 common mistakes examiners make in digital forensics and how to avoid them.

40 Min episode Podcast

How to use the Keychain in Physical Analyzer

The blog reviews what the iOS Keychain is, how to obtain it, and how the forensic tool should leverage it to aid in the decryption of secure applications.

5 min read Read Blog

Hands-on EXERCISE:

Let's examine
the test data



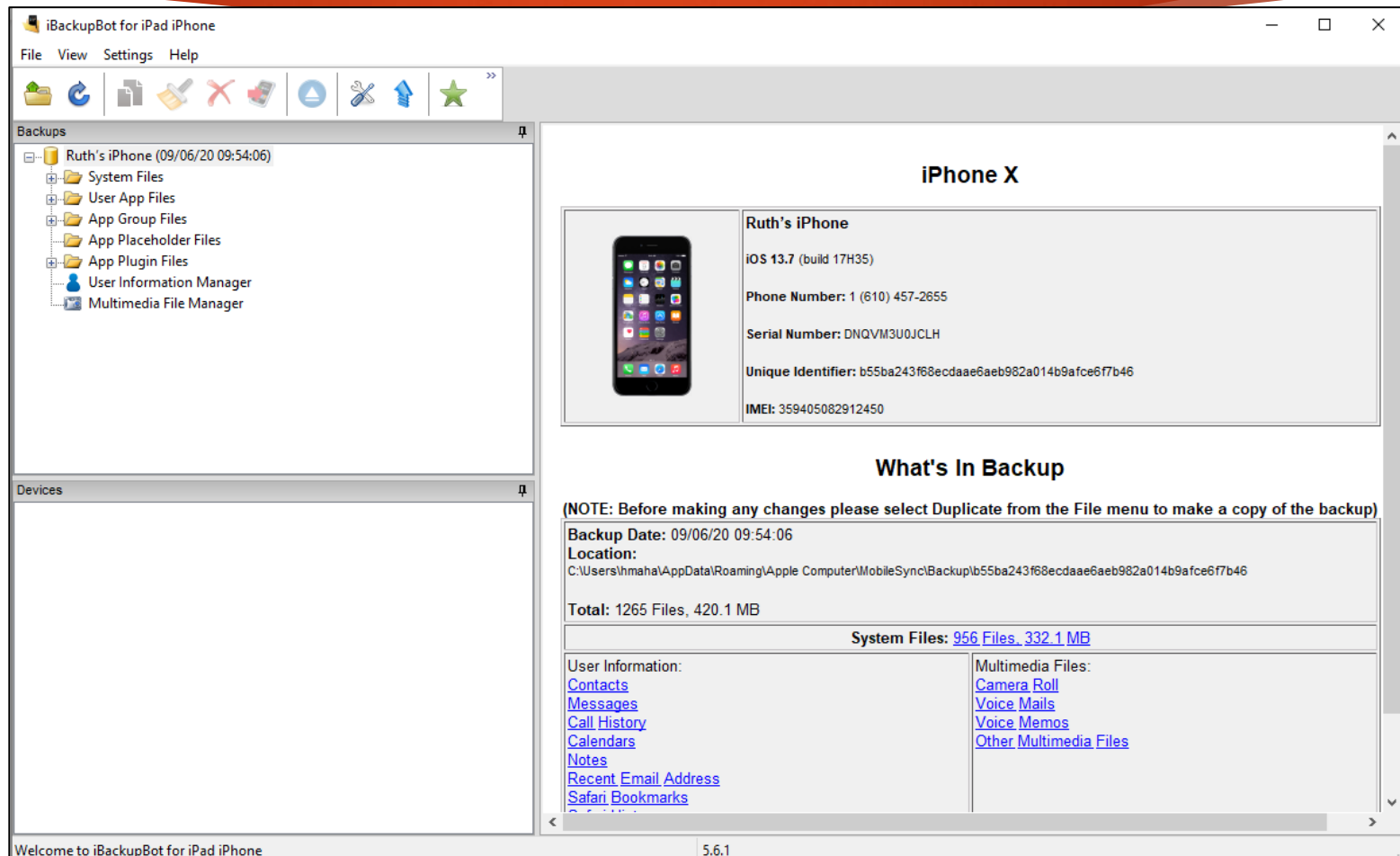
Examining the Data

You will need

- ▶ The lab document from Dropbox – **for585.com/techtues**
 - ▶ Download and install iBackupBot (iCopyBot)
 - ▶ If on Windows, Agent Ransack and be downloaded and installed (optional)
 - ▶ Download and install plist editor
 - ▶ Download the lab data from Dropbox

Caveat – To use free tools, we cannot encrypt the backup during the iTunes extraction – this means you will NOT have call logs in the backup.

iBackupBot Demo





Finding the Test Data

When the timestamps don't make sense...

The screenshot shows an iPhone backup interface with tabs for Contacts, Messages, Call History, Calendar, Notes, Recent Email, Safari Bookmarks, and Safari History. The Messages tab is selected, showing a list of messages. Two messages are highlighted with red boxes:

- Message 1: 06/05/18 07:22:40 From: +17029106555 Read at: 08/27/09 17:20:00 Huh
- Message 2: 10/09/68 19:29:52 To: +17029106555 Testing time change on iOS

The contact list on the left shows two entries for 'Orlando[+170291...]' with a count of 2 and a last timestamp of 12/31/00 19:00:00.

Parsing SMS.db

DB Browser for SQLite - C:\Users\hmaha\Desktop\SANS_TechTues_data\sms.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 Pragas Execute SQL

SQL 1

```
29 datetime(chat.last_read_message_timestamp + 978307200, 'unixepoch', 'localtime')
30 else 'NA'
31 END as "Last Read",
32 attachment.filename,
33 datetime(attachment.created_date+978307200, 'unixepoch', 'localtime') AS "Attachment Date",
34 attachment.mime_type,
35 attachment.total_bytes
36 FROM message
37 left join chat_message_join on chat_message_join.message_id=message.ROWID
38 left join chat on chat.ROWID=chat_message_join.chat_id
39 left join attachment on attachment.ROWID=chat_message_join.chat_id
40 order by message.date_read desc;
```

	ROWID	chat_id	handle_id	text	service	account	account_login	chat_identifier
67	53	4	4	Curie basket new up!	SMS	e:	E:	+14708001223
68	55	4	4	Test	SMS	e:	E:	+14708001223
69	56	4	4	Test for r	SMS	e:	E:	+14708001223
70	59	4	4	Testing something. I need for you to keep this safe	SMS	e:	E:	+14708001223
71	64	4	4	Loved "Hey. Thanks for sending me that - looks like we can ...	SMS	e:	E:	+14708001223
72	65	4	4	Thank you! I promise I'm working for you. Not against you	SMS	e:	E:	+14708001223
73	66	4	4	And same goes for JM	SMS	e:	E:	+14708001223
74	69	4	4	Yes. That's the plan. Wade mentioned a good meetup for me ...	SMS	e:	E:	+14708001223
75	71	4	4	I would love to work my way up	SMS	e:	E:	+14708001223
76	73	4	4	Okay	SMS	e:	E:	+14708001223
77	74	10	10	Ruth you're eligible for \$417.11...	SMS	e:	E:	+15033696546
78	77	11	11	Testing time change on iOS	SMS	e:	E:	+17029106555
79	79	1	1	Incoming text with time change	iMessage	p:+16104572655	E:ruthlessg1rl11@gmail.com	+17034241981
80	80	12	12	Who's this ?	iMessage	p:+16104572655	E:ruthlessg1rl11@gmail.com	+17029106555
81	81	12	12	Who's this	iMessage	p:+16104572655	E:ruthlessg1rl11@gmail.com	+17029106555

Parsing Call_History.storedata

****If you were able to encrypt the backup and decrypt it!**

```
select
  z_pk AS "Call Sequence #",
  zaddress AS "Phone Number",
  zduration AS "Call in Seconds",
  case
    when zoriginated = 0 then "Incoming"
    when zoriginated = 1 then "Outgoing"
  end AS "Call Direction",
  case
    when zanswered = 0 then "Call Missed"
    when zanswered = 1 then "Call Answered"
  end as "Call Status",
  datetime(zdate+978307200,'unixepoch','localtime') AS
  "Timestamp"
from zcallrecord
```

Methods for Obtaining a FFS

- ▶ Cellebrite UFED using checkm8 extraction
 - ▶ Temporary jailbreak that runs in memory
 - ▶ Does not permanently change the device
- ▶ Checkra1n jailbreak using a Mac or Linux system
 - ▶ Permanently changes the device
- ▶ Elcomsoft – commercial tool offering checkra1n support

Caveat – read up on it and test before you try a public jailbreak on live evidence!

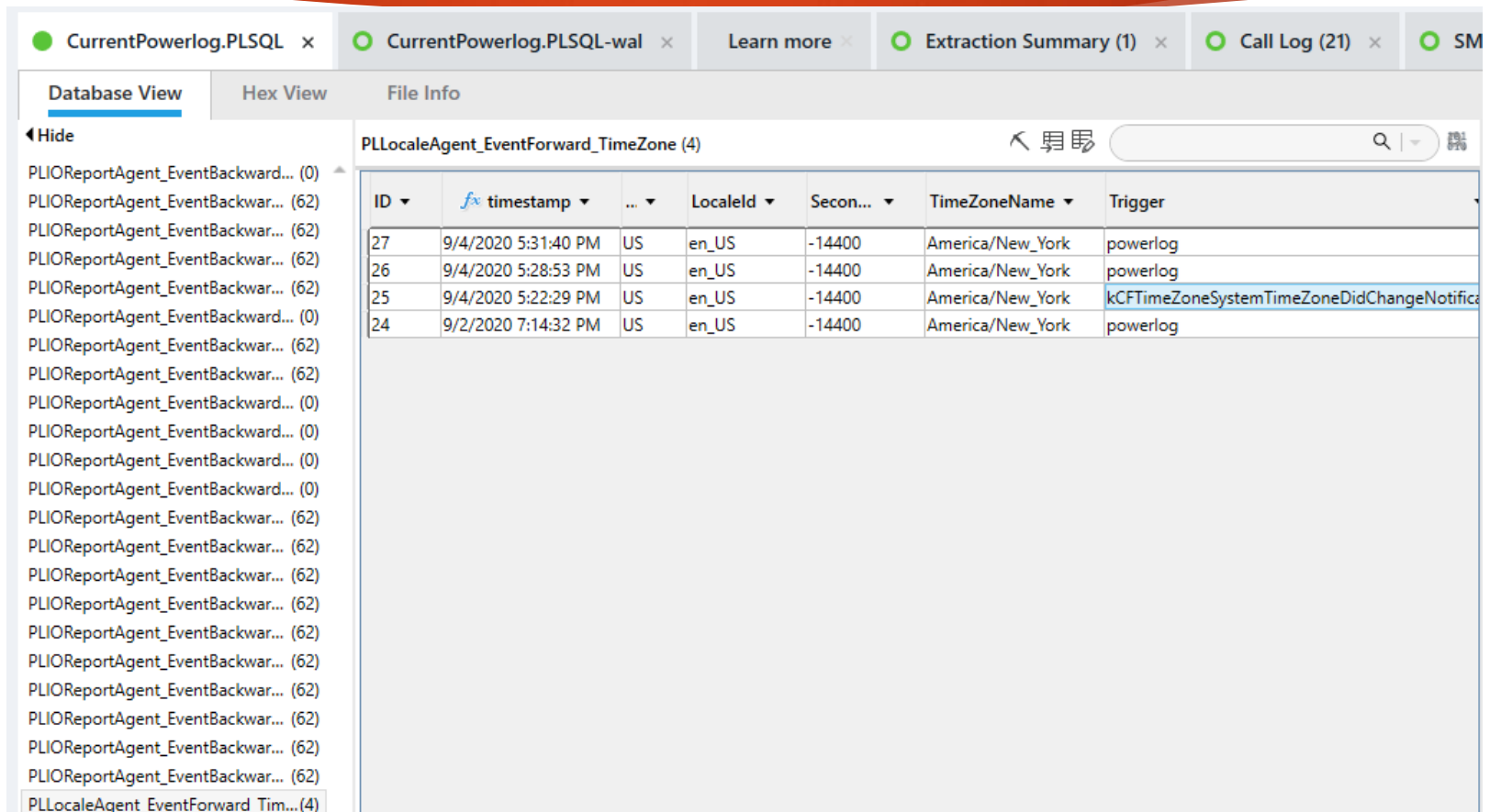


Reviewing the Results(1)

Parties	Timestamp	Duration	Status	Country code
From: +17029106555 Orlando	9/1/2020 4:05:18 PM(UTC+0)	00:00:08	Answered	us
From: +17029106555 Orlando	9/1/2020 4:03:42 PM(UTC+0)	00:00:00	Rejected	us
From: +17029106555 Orlando	9/1/2020 4:03:26 PM(UTC+0)	00:00:00	Rejected	us
To: 7032226411	9/1/2020 4:01:37 PM(UTC+0)	00:00:00	Not answered	us
To: 7034241981	9/1/2020 4:00:51 PM(UTC+0)	00:00:20	Answered	us
From: +16104578752 Scam Likely	8/20/2020 7:33:43 PM(UTC+0)	00:00:00	Missed	us

Timestamp	Read	Folder	Parties	Body	Status
9/4/2020 12:04:41 PM(UTC+0)	9/1/2020 4:03:20 PM(UTC+0)	Inbox	From: 456 456 (owner) To:	Reminder: Please refill your plan within the next 3 days to ensu...	Read
9/4/2020 12:04:41 PM(UTC+0)		Recents	From: 456 456		
9/2/2020 12:03:57 PM(UTC+0)	9/1/2020 4:03:20 PM(UTC+0)	Inbox	From: 456 456 (owner) To:	Reminder: Please refill your plan within the next 3 days to ensu...	Read
9/1/2020 4:03:00 PM(UTC+0)	9/1/2020 4:03:12 PM(UTC+0)	Inbox	From: +17029106555 Orlando (owner) To:	Huh	Read
9/1/2020 4:02:35 PM(UTC+0)		Sent	From: (owner) To: +17029106555 Orlando	Testing time change on iOS	Sent

Reviewing the Results(2)



The screenshot shows a database application window with multiple tabs at the top: 'CurrentPowerlog.PLSQL', 'CurrentPowerlog.PLSQL-wal', 'Learn more', 'Extraction Summary (1)', 'Call Log (21)', and 'SM'. The 'Database View' tab is active, displaying a tree view on the left and a table of results on the right. The tree view lists various event logs, with 'PLLocaleAgent_EventForward_Tim... (4)' selected. The table on the right, titled 'PLLocaleAgent_EventForward_TimeZone (4)', contains the following data:

ID	timestamp	...	LocaleId	Secon...	TimeZoneName	Trigger
27	9/4/2020 5:31:40 PM	US	en_US	-14400	America/New_York	powerlog
26	9/4/2020 5:28:53 PM	US	en_US	-14400	America/New_York	powerlog
25	9/4/2020 5:22:29 PM	US	en_US	-14400	America/New_York	kCFTimeZoneSystemTimeZoneDidChangeNotifica
24	9/2/2020 7:14:32 PM	US	en_US	-14400	America/New_York	powerlog

Summary of this Test

- ▶ Time alteration on an iOS device will show activities reflecting the “altered timestamp”
- ▶ The currentpowerlog will show the timechange – need a FFS extraction for this
- ▶ Testing for this type of evidence tampering did not take long. You just need to know where to start!

Reality:

- ▶ This test required an encrypted backup to examine the call_history.stored data
- ▶ This test required a FFS to get the CurrentPowerlog.PLSQL

References and Resources

- ▶ FOR585.com.course
- ▶ <https://www.cellebrite.com/en/blog/if-i-could-turn-back-time-a-closer-look-at-ios-time-modifications/>
- ▶ www.smarterforensics.com/blog
- ▶ Stay tuned for more blogs on Cellebrite's site, Smarterforensics and DFIR Review

FOR585 Advanced Smartphone Forensics



- ▶ Course launched in 2014
- ▶ GASF Cert – Vendor neutral available to everyone
- ▶ Co-authored by Heather Mahalik and Lee Crognale
Addresses the hardest to tackle topics (Encryption, Parsing, Query drafting, decompiling malware, etc.)
- ▶ Primary focus is analysis and understanding the artifacts
- ▶ Includes 30+ hands-on labs + 1 capstone challenge + 1 take home case
- ▶ Is vendor NEUTRAL – We teach you the best methods, not how to use commercial tools
- ▶ 50% LE discount available for several seats (online and live training)
- ▶ 50% discount for alumni!

FOR585 Training Opportunities

- ▶ Live Online
 - ▶ March 22– March 27 (SANS 2021)
 - ▶ 9AM EDT – 5PM EDT
 - ▶ April 26 – May 1
 - ▶ 3AM EDT – 11AM EDT
 - ▶ May 3 – May 8
 - ▶ 9AM EDT – 5PM EDT
- ▶ On Demand
 - ▶ Train anytime anywhere according to your schedule



QUESTIONS?

Heather Mahalik

hmahalik@gmail.com

Twitter: @hmahalik



Lee Crognale

domenica.crognale@gmail.com

Twitter: @domenicacrognal