

# The iOS of Sauron: How iOS Tracks Everything You Do

Sarah Edwards | oompa@csh.rit.edu | @iamevtwin | mac4n6.com

# PATTERN OF LIFE

Application Usage

Device State

Data Usage

Health

Location

# INFO & CAVEATS

## Jailbroken Device

- Files discussed are not available without root/physical access to the device.
  - Except Health databases if backup is encrypted.
- Capture files while screen unlocked. (Some are encrypted when device is locked)

## Timestamps - Never Trust Implicitly

- Different Storage Formats
- Some differ from actual event time (Always, always, always test before you report wrong data!)

## Data Retention

- Each database/file is different, each table in the database is different
- iOS versions may adjust these.
- Data Retentions in this presentation are what I see in my own data - yours may differ.

## Work in Progress

- I reserve the right to change update the details of this presentation to reflect new research.
- Find updated copies of this presentation on [mac4n6.com](http://mac4n6.com)

## Test Devices

- iPhone 6 (iOS 9.0.2), Apple Watch 1 (watchOS 2.0)

# DATA FILES FILE PATH & DETAILS

- **CoreDuet** - /private/var/mobile/Library/CoreDuet/
  - **coreduet.db** (31 Tables)
  - coreduetClassA.db (31 Tables)
  - **coreduetClassD.db** (31 Tables)
  - Knowledge/knowledgeC.db (5 Tables)
  - People/interactionC.db (9 Tables)
- **Aggregate Dictionary** - /private/var/mobile/Library/AggregateDictionary/
  - ADDataStore.sqlitedb ( 4 Tables)
- **Battery Life (PowerLog)** - /private/var/mobile/Library/BatteryLife/
  - **CurrentPowerlog.PLSQL** (257 Tables)
  - Archives/powerlog\_YYYY-MM-DD\_XXXXXXXX.PLSQL.gz (Previous ~5 Days)

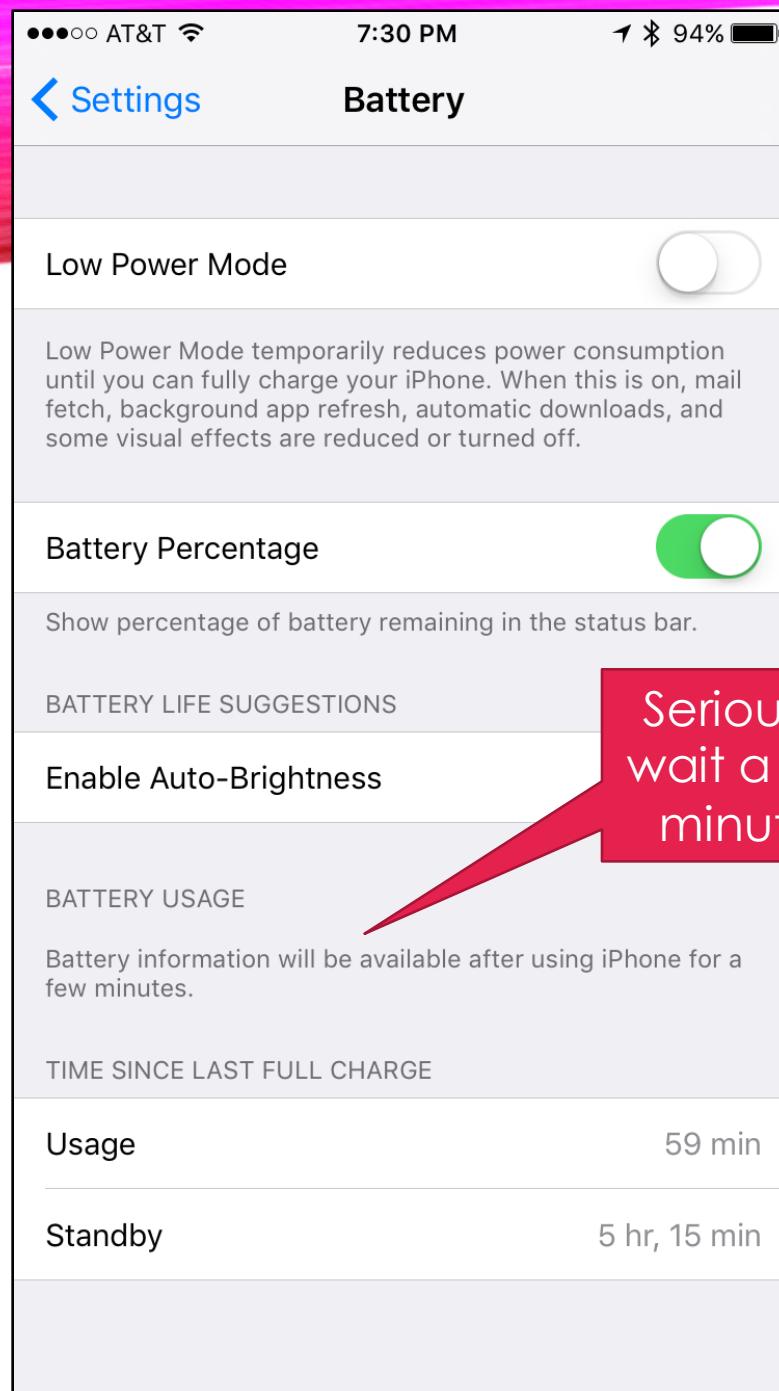
Note:  
Files highlighted are discussed in this presentation.

# DATA FILES FILE PATH & DETAILS

- **networkd** - /private/var/networkd/
  - netusage.sqlite (13 Tables)
- **Health** - /private/var/mobile/Library/Health/
  - **healthdb.sqlite** (11 Tables)
  - **healthdb\_secure.sqlite** (16 Tables)
- **routined** - /private/var/mobile/Library/Caches/com.apple.routined/
  - **cache\_encryptedB.db** (5 Tables)
  - **StateModel1.archive**
  - **StateModel2.archive**
- **locationd** - /private/var/root/Library/Caches/locationd/
  - **cache\_encryptedA.db** (79 Tables)
  - **lockCache\_encryptedA.db** (51 Tables)
  - **cache\_encryptedB.db** (167 Tables)
  - cache\_encryptedC.db (9 Tables)

# APPLICATION USAGE





# APP BATTERY USAGE TIME

BATTERY USAGE		
Last 24 Hours	Last 7 Days	(
Mail Background Activity	27%	
Twitter	23%	
Settings	17%	
Facebook	13%	
TripIt	11%	
Home & Lock Screen	3%	
No Cell Coverage	2%	
Camera	2%	
The Weather Background Activity	1%	
Siri	1%	

Shows proportion of battery used by each app in the last 24 hours.

TIME SINCE LAST FULL CHARGE

BATTERY USAGE		
Last 24 Hours	Last 7 Days	(
Twitter 1.9 hrs on screen – 1 min background	23%	
Mail 12 min on screen – 1.3 hrs background	18%	
Health 50 min on screen	13%	
Activity 52 min on screen	12%	
Facebook 16 min on screen – 25 min background	9%	
Wallet 1 min on screen	4%	
Home & Lock Screen 12 min on screen	3%	
TripIt 1 min on screen	2%	
OverDrive 6 min on screen	2%	
Calculator 9 min on screen	2%	
Messages 3 min on screen	2%	
Safari 6 min on screen	2%	

# APP USAGE TIME

## CurrentPowerlog.PLSQL

```
1 select datetime(timestamp,'unixepoch','utc') as timestamp,  
2 BackgroundTime,  
3 ScreenOnTime,BundleID  
4 from PLAppTimeService_Aggregate_AppRunTime
```

	timestamp	BackgroundTime	ScreenOnTime	BundleID
1245	2016-04-02 17:00:00	2312.881339	0.0	com.apple.SafariViewService
1246	2016-04-02 17:00:00	22.20416	0.0	com.apple.mobilemail
1247	2016-04-02 18:00:00	173.04662	0.0	net.whatsapp.WhatsApp
1248	2016-04-02 18:00:00	4064.636366	0.0	com.apple.SafariViewService
1249	2016-04-02 18:00:00	8.985416	0.0	com.facebook.Facebook
1250	2016-04-02 18:00:00	0.0	1.297704	com.apple.lock-screen
1251	2016-04-02 18:00:00	12.598638	2.51937	com.atebits.Tweetie2
1252	2016-04-02 18:00:00	0.0	9.717909	com.apple.springboard.home-screen
1253	2016-04-02 18:00:00	0.0	25.769507	com.apple.MobileSMS

### Data Retention:

~2 Weeks

(Don't Forget  
Powerlog Archives!)

### Timestamps:

Accurate  
Per Day, Per Hour

### Timing:

In Seconds

# APP USAGE - BY TIME

## CurrentPowerlog.PLSQL

```

1 select BundleID, sum(ScreenOnTime) as ScreenTime
2 from PLAppTimeService_Aggregate_AppRunTime
3 group by BundleID
4 order by ScreenTime desc
5 limit 10

```

	BundleID	ScreenTime
1	com.atebits.Tweetie2	20319.951987
2	com.overdrive.odm	11546.506357
3	com.apple.mobilemail	5683.870588
4	com.apple.MobileSMS	3649.106116
5	com.facebook.Facebook	3097.038284
6	com.apple.mobilesafari	2561.764628
7	com.apple.Music	2195.374459
8	com.united.UnitedCustomerFacingiPhone	1895.538196
9	com.apple.lock-screen	1683.959374
10	com.apple.springboard.home-screen	1615.986644

```

1 select BundleID, sum(BackgroundTime) as BackgroundTime
2 from PLAppTimeService_Aggregate_AppRunTime
3 group by BundleID
4 order by BackgroundTime desc
5 limit 10

```

	BundleID	BackgroundTime
1	com.apple.SafariViewService	204951.559755
2	com.apple.quicklook.quicklookd	38882.607182
3	com.overdrive.odm	36660.672928
4	com.apple.mobilemail	15171.125882
5	com.apple.Music	9480.383636
6	net.whatsapp.WhatsApp	5279.943346
7	com.glympse.iphone.glympse	4459.481422
8	com.facebook.Messenger	3736.228571
9	com.facebook.Facebook	2603.756805
10	com.atebits.Tweetie2	1481.23503

# APP USAGE - BY HOUR OR BY APP

## CurrentPowerlog.PLSQL

```

1 select datetime(timestamp,'unixepoch','localtime') as "Day_Hour",
2      BundleID, ScreenOnTime
3      from PLAppTimeService_Aggregate_AppRunTime
4      where Day_Hour like '2016-04-10 10:00:00'
5      order by ScreenOnTime desc

```

	Day_Hour	BundleID	ScreenOnTime
1	2016-04-10 10:00:00	com.atebits.Tweetie2	470.419365
2	2016-04-10 10:00:00	com.apple.springboard.home-screen	8.13034
3	2016-04-10 10:00:00	com.apple.mobilemail	5.144626
4	2016-04-10 10:00:00	com.apple.control-center	2.324725
5	2016-04-10 10:00:00	com.apple.camera	2.108956
6	2016-04-10 10:00:00	com.apple.lock-screen	1.104772
7	2016-04-10 10:00:00	com.myfitnesspal.mfp	0.0
8	2016-04-10 10:00:00	com.audible.iphone	0.0
9	2016-04-10 10:00:00	com.apple.WebKit.WebContent	0.0
10	2016-04-10 10:00:00	com.apple.WebKit.Networking	0.0
11	2016-04-10 10:00:00	com.facebook.Messenger	0.0
12	2016-04-10 10:00:00	net.whatsapp.WhatsApp	0.0
13	2016-04-10 10:00:00	com.facebook.Facebook	0.0

```

1 select datetime(timestamp,'unixepoch','localtime') as "Day_Hour",
2      BundleID, ScreenOnTime
3      from PLAppTimeService_Aggregate_AppRunTime
4      where BundleID like '%tweetie%' and ScreenOnTime > 0
5      order by Day_Hour

```

	Day_Hour	BundleID	ScreenOnTime
12	2016-04-02 12:00:00	com.atebits.Tweetie2	114.810658
13	2016-04-02 23:00:00	com.atebits.Tweetie2	810.924827
14	2016-04-03 03:00:00	com.atebits.Tweetie2	1259.868143
15	2016-04-03 04:00:00	com.atebits.Tweetie2	130.449583
16	2016-04-04 18:00:00	com.atebits.Tweetie2	767.512703
17	2016-04-04 21:00:00	com.atebits.Tweetie2	3.264545
18	2016-04-06 23:00:00	com.atebits.Tweetie2	27.539372
19	2016-04-07 19:00:00	com.atebits.Tweetie2	14.098393

```

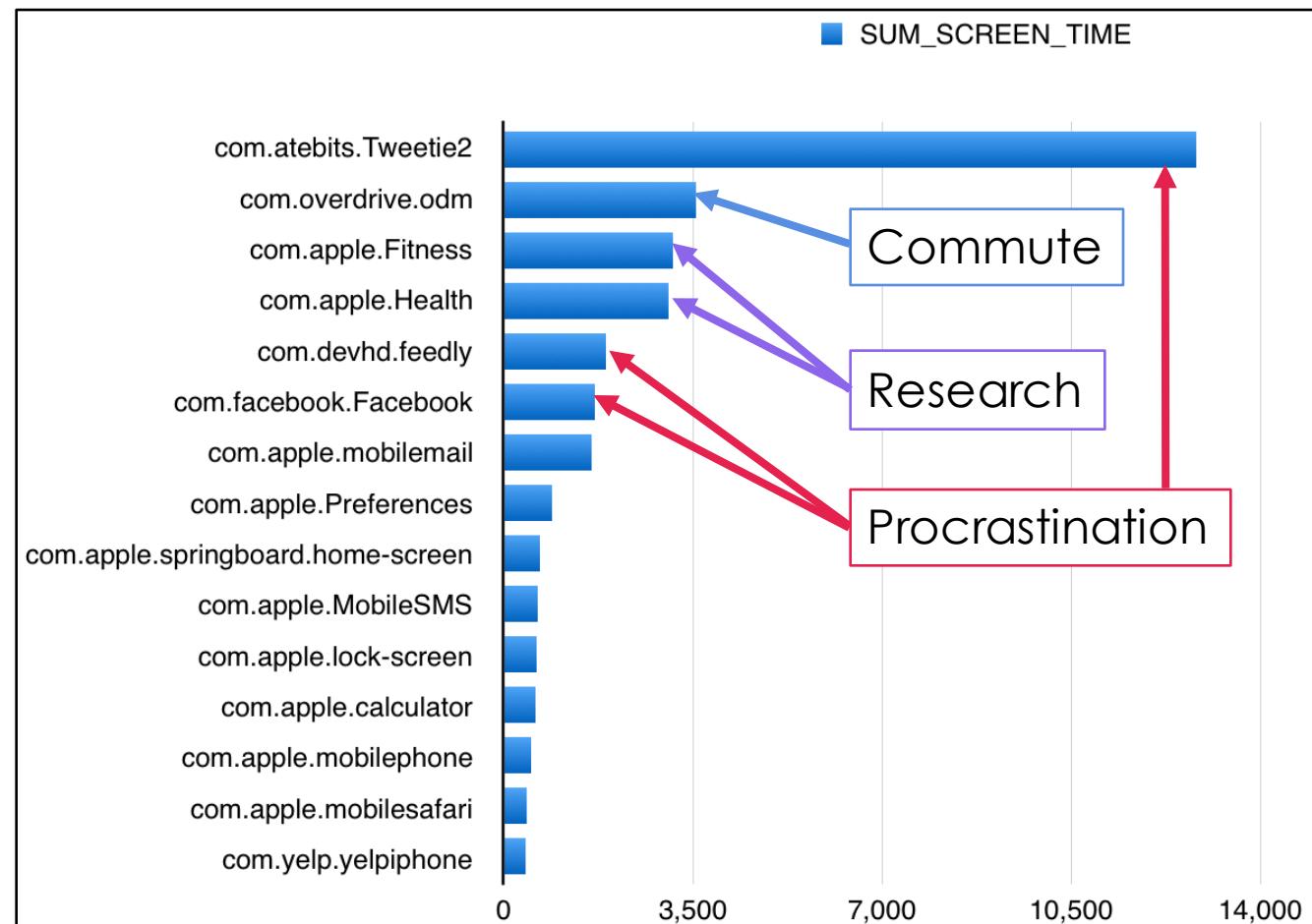
1 select BundleID, sum(ScreenOnTime) as SUM_SCREEN_TIME
2 from PLAppTimeService_Aggregate_AppRunTime
3 group by BundleID
4 order by SUM_SCREEN_TIME desc

```

	BundleID	SUM_SCREEN_TIME
1	com.atebits.Tweetie2	12808.434844
2	com.overdrive.odm	3562.569724
3	com.apple.Fitness	3127.182036
4	com.apple.Health	3045.792654
5	com.devhd.feedly	1897.791861
6	com.facebook.Facebook	1689.233466
7	com.apple.mobilemail	1619.268382
8	com.apple.Preferences	905.630977
9	com.apple.springboard.home-screen	675.553936
10	com.apple.MobileSMS	643.05514
11	com.apple.lock-screen	616.998478
12	com.apple.calculator	591.707975
13	com.apple.mobilephone	518.574817
14	com.apple.mobilesafari	428.364785
15	com.yelp.yelpiphone	417.19901

# OVERALL APP SCREEN TIME

## CurrentPowerlog.PLSQL



# APP USAGE PLAY-BY-PLAY

## CurrentPowerlog.PLSQL

```
1 select datetime(timestamp,"unixepoch","localtime") as LOCAL_TIME, BundleID  
2 from PLScreenStateAgent_EventForward_ScreenState
```

	ts	bundleID
78	2016-04-11 19:48:21	com.apple.mobletimer
79	2016-04-11 19:48:23	com.apple.springboard.home-screen
80	2016-04-11 19:49:20	com.atebits.Tweetie2
81	2016-04-11 19:50:00	com.apple.springboard.home-screen
82	2016-04-11 19:50:21	com.linkedin.LinkedIn
83	2016-04-11 19:51:04	com.apple.springboard.home-screen
84	2016-04-11 19:51:21	net.whatsapp.WhatsApp
85	2016-04-11 19:51:41	com.apple.springboard.home-screen
86	2016-04-11 19:52:20	com.apple.MobileSMS
87	2016-04-11 19:52:42	com.apple.springboard.home-screen
88	2016-04-11 19:53:20	com.squarespace.metrics
89	2016-04-11 19:54:13	com.apple.springboard.home-screen
90	2016-04-11 19:54:20	com.wunderground.weatherunderground

**Data Retention:**  
~1 Day

**Timestamp Warning:**  
My timestamps were  
off by +9:20



**Timing:**  
In Seconds

# DEVICE STATE



# DEVICE LOCK STATE

## coreduetd.db

```
1 select ZCREATIONDATE AS MAC_EPOCH,  
2      datetime(ZCREATIONDATE+978307200,'unixepoch') as CREATETIME_UTC,  
3      time(ZLOCALTIME,'unixepoch') as LOCAL_DEVICE_TIME,  
4      time(ZCREATIONDATE-ZLOCALTIME,'unixepoch') as TIMEZONE,  
5      ZLOCKSTATE --0=Unlocked, 1=Locked  
6   from ZCDDMSCREENLOCKEVENT
```

	MAC_EPOCH	CREATETIME_UTC	LOCAL_DEVICE_TIME	TIMEZONE	ZLOCKSTATE
325	479753024	2016-03-15 16:43:44	10:43:44	06:00:00	0
326	479753088	2016-03-15 16:44:48	10:44:48	06:00:00	1
327	479761408	2016-03-15 19:03:28	13:03:28	06:00:00	0
328	479761600	2016-03-15 19:06:40	15:06:40	04:00:00	1
329	479761728	2016-03-15 19:08:48	15:08:48	04:00:00	0
330	479761984	2016-03-15 19:13:04	15:13:04	04:00:00	1
331	479762496	2016-03-15 19:21:36	15:21:36	04:00:00	0

**Data Retention:**

~1 Month

**Time Zone:**

"ZCREATIONDATE" -  
"ZLOCALTIME"

**Timestamp Warning:**

Timestamps seem to be off ever so slightly.  
(~1m or so)



**State:**

0 = Unlocked

1 = Locked

# AIRPLANE MODE

## coreduetd.db

```
1 select ZCREATIONDATE AS MAC_EPOCH,  
2      datetime(ZCREATIONDATE+978307200,'unixepoch') as CREATETIME_UTC,  
3      time(ZLOCALTIME,'unixepoch') as LOCAL_DEVICE_TIME,  
4      time(ZCREATIONDATE-ZLOCALTIME,'unixepoch') as TIMEZONE,  
5      ZAIRPLANEMODEON --0=Off, 1=On  
6  from ZCDDMAIRPLANEMODEEVENT
```

	MAC_EPOCH	CREATETIME_UTC	LOCAL_DEVICE_TIME	TIMEZONE	ZAIRPLANEMODEON
14	479745792	2016-03-15 14:43:12	07:43:12	07:00:00	0
15	479745792	2016-03-15 14:43:12	07:43:12	07:00:00	0
16	479749440	2016-03-15 15:44:00	09:44:00	06:00:00	1
17	479749632	2016-03-15 15:47:12	09:47:12	06:00:00	0
18	479749632	2016-03-15 15:47:12	09:47:12	06:00:00	0
19	479749696	2016-03-15 15:48:16	09:48:16	06:00:00	1
20	479749696	2016-03-15 15:48:16	09:48:16	06:00:00	1
21	479761408	2016-03-15 19:03:28	13:03:28	06:00:00	0
22	480170752	2016-03-20 12:45:52	08:45:52	04:00:00	1
23	480178432	2016-03-20 14:53:52	10:53:52	04:00:00	0

### Data Retention:

~1 Month

### Timestamp Warning:

Timestamps seem to be off ever so slightly. (~1m)

Also may get multiple timestamp hits when duration is short between state changes



**0 = Airplane Mode Off**

**1 = Airplane Mode On**

# DEVICE BATTERY LEVEL

## CurrentPowerlog.PLSQL

```
1 select datetime(timestamp,'unixepoch','localtime') as LOCAL_TIME,  
2 Level, RawLevel  
3 from PLBatteryAgent_EventBackward_Battery  
4
```

	LOCAL_TIME	Level	RawLevel
1012	2016-04-09 18:51:25	78.0	73.4960272417707
1013	2016-04-09 18:51:46	77.0	73.3257661748014
1014	2016-04-09 18:52:06	77.0	73.2690124858116
1015	2016-04-09 18:52:26	77.0	73.155505107832
1016	2016-04-09 18:52:47	77.0	73.0987514188422
1017	2016-04-09 18:53:07	77.0	72.9852440408627
1018	2016-04-09 18:53:28	77.0	72.9284903518729
1019	2016-04-09 18:53:48	77.0	72.8717366628831

**Battery Level:**  
Every ~20 Seconds

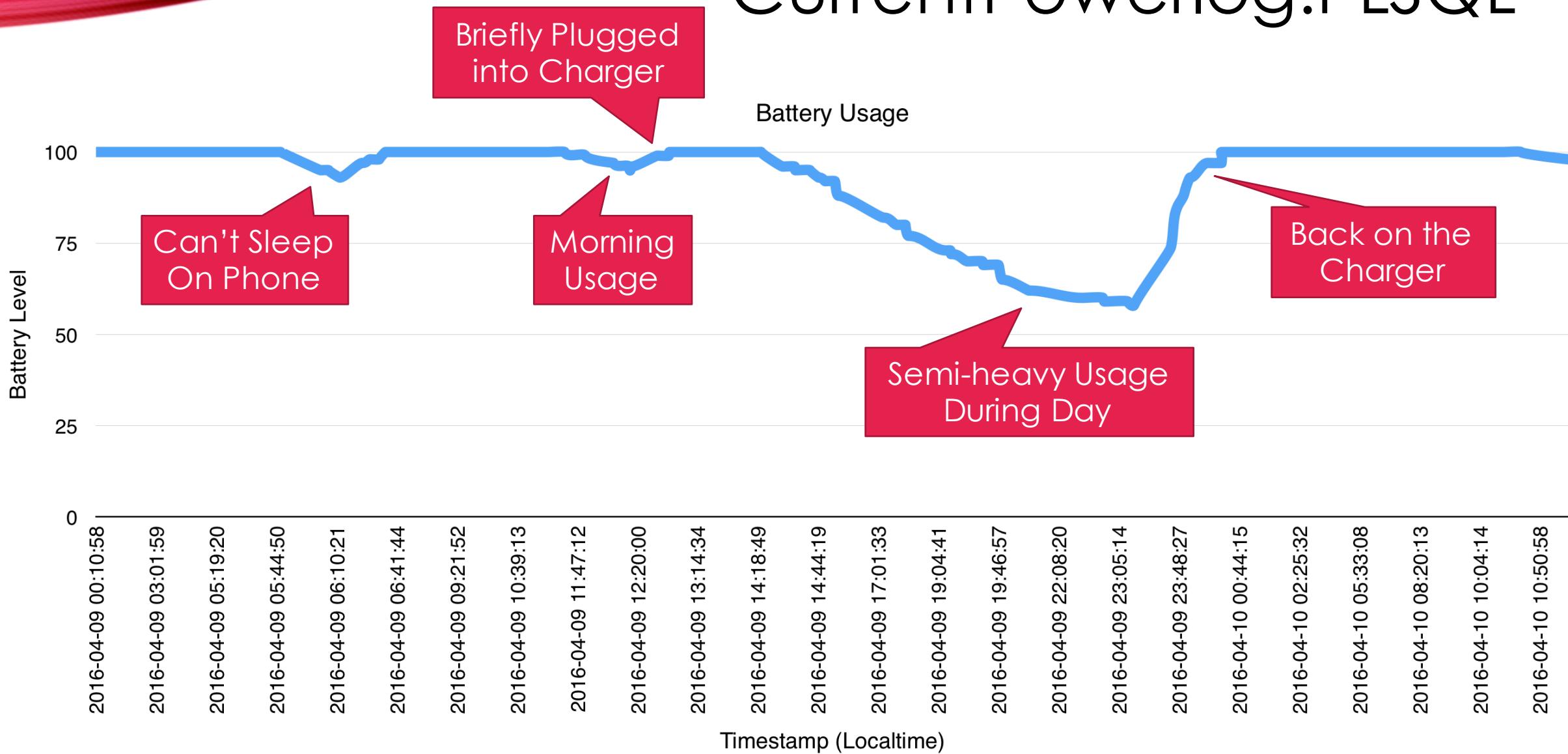
**Data Retention:**  
~2 Days

**Timestamp Warning:**  
Mine were off by about  
+6 minutes

**Note:**  
Level Vs. RawLevel

# DEVICE BATTERY LEVEL

## CurrentPowerlog.PLSQL



# DEVICE PLUGIN STATE

## coreduetdClassD.db

```
1 SELECT datetime(ZCREATIONDATE+978307200,'unixepoch', 'LOCALTIME') as PLUG_TIME,  
2 time(ZCREATIONDATE-ZLOCALTIME,'unixepoch') as TIMEZONE, ZCABLESTATE  
3 FROM ZCDDMPLUGINEVENT  
4 where PLUG_TIME like '2016-04-11%'
```

	PLUG_TIME	TIMEZONE	ZCABLESTATE	
1	2016-04-11 07:42:56	04:00:00	0	Unplugged from bedside charger
2	2016-04-11 07:52:32	04:00:00	1	Commute Time - Car Charger
3	2016-04-11 08:37:20	04:00:00	0	Finally at Work!
4	2016-04-11 17:43:28	04:00:00	1	Long commute home..
5	2016-04-11 17:43:28	04:00:00	0	...sometimes it doesn't take...
6	2016-04-11 17:43:28	04:00:00	1	...and you get multiple entries.
7	2016-04-11 18:30:24	04:00:00	0	Made good time home!
8	2016-04-11 19:46:08	04:00:00	1	Research time!

**Data Retention:**

~1 Month

**Timestamps:**

Accurate

**State:**

0 = Unplugged

1 = Plugged In

# DATA USAGE



# APP/PROCESS DATA USAGE

## CurrentPowerlog.PLSQL

```
1 select datetime(timestamp,'unixepoch','localtime') as Timestamp, datetime(timestampend,'unixepoch','localtime') as TimestampEnd,
2 BundleName, ProcessName, cellin, CellOut, WiFiIn, WiFiOut from PLProcessNetworkAgent_EventInterval_UsageDiff
3
4
```

	Timestamp	TimestampEnd	BundleName	ProcessName	CellIn	CellOut	WiFiIn	WiFiOut	
243	2016-04-12 08:05:10	2016-04-12 08:35:11	NULL	locationd	0	0	114433	35275	
244	2016-04-12 08:05:10	2016-04-12 08:35:11	NULL	configd	0	0	334	328	
245	2016-04-12 08:05:10	2016-04-12 08:35:11	com.audible.iphone	com.audible.iphone	0	0	30264	32335	
246	2016-04-12 08:35:11	2016-04-12 09:05:16	com.apple.mobilenotes	com.apple.mobilenotes	20737	4994	0	0	
247	2016-04-12 08:35:11	2016-04-12 09:05:16	com.apple.mobilemail	com.apple.mobilemail	353911	19967	0	0	
248	2016-04-12 08:35:11	2016-04-12 09:05:16	NULL	itunesstored	12491	7707	0	0	
249	2016-04-12 08:35:11	2016-04-12 09:05:16	NULL	CommCenter	2855	6414	0	0	
250	2016-04-12 08:35:11	2016-04-12 09:05:16	NULL	SpringBoard	103	55	0	0	

**Data Retention:**  
~2 Days

**Timestamps:**  
Accurate

**Unit:**  
Bytes (I believe)

# APP/PROCESS DATA USAGE

## CurrentPowerlog.PLSQL

```

1 select BundleName, ProcessName, sum(cellin) as CellIn, sum(cellout) as CellOut,
2 sum(wifiin) as WifiIn, sum(wifiout) as WifiOut
3 from PLProcessNetworkAgent_EventInterval_UsageDiff
4 group by ProcessName
5 order by CellIn desc

```

	BundleName	ProcessName	CellIn	CellOut	WifiIn	WifiOut
1	com.atebits.Tweetie2	com.atebits.Tweetie2	89216717	3671288	3695493	138472
2	com.apple.mobilemail	com.apple.mobilemail	37196932	1123721	9391350	430627
3	com.safariflow.SafariQueue	com.safariflow.SafariQueue	19692790	452177	0	0
4	com.facebook.Facebook	com.facebook.Facebook	11405436	578179	72674	31171
5	com.redfin.redfin	com.redfin.redfin	5448825	456291	0	0
6	NULL	syncdefaultsd	2754810	1070054	357067	144049
7	com.tripit.tripitmobile.paid	com.tripit.tripitmobile.paid	2125361	239751	0	0
8	NULL	itunesstored	1442274	310128	1245377	97541
9	com.apple.mobilenotes	com.apple.mobilenotes	970098	313980	755589	229989
10	NULL	searchd	814132	91134	187091	27422

```

1 select BundleName, ProcessName, sum(cellin) as CellIn, sum(cellout) as CellOut,
2 sum(wifiin) as WifiIn, sum(wifiout) as WifiOut
3 from PLProcessNetworkAgent_EventInterval_UsageDiff
4 group by ProcessName
5 order by WifiIn desc

```

	BundleName	ProcessName	CellIn	CellOut	WifiIn	WifiOut
1	com.apple.AppStore	com.apple.AppStore	195091	27220	212536708	4969
2	com.apple.mobilemail	com.apple.mobilemail	37196932	1123721	9391350	430627
3	com.atebits.Tweetie2	com.atebits.Tweetie2	89216717	3671288	3695493	138472
4	NULL	nsurlsessiond	10957	5988	3238824	143778987
5	NULL	locationd	120121	13250	2399528	788435
6	NULL	itunesstored	1442274	310128	1245377	97541
7	com.apple.mobilenotes	com.apple.mobilenotes	970098	313980	755589	229989
8	NULL	syncdefaultsd	2754810	1070054	357067	144049
9	com.apple.WebSheet	com.apple.WebSheet	0	0	233939	8192
10	NULL	assistantd	157596	168179	207062	149994

# APP/PROCESS DATA USAGE

## CurrentPowerlog.PLSQL

```
1 select datetime(timestamp,'unixepoch','localtime') as Timestamp, datetime(timestampend,'unixepoch','localtime') as EndTimestamp,
2 BundleName, ProcessName, CellIn, CellOut, WifiIn, WifiOut from PLProcessNetworkAgent_EventInterval_UsageDiff
3 where ProcessName like '%facebook.facebook%'
```

	Timestamp	EndTimestamp	BundleName	ProcessName	CellIn	CellOut	WifiIn	WifiOut	
1	2016-04-01 02:20:58	2016-04-01 02:53:59	com.facebook.Facebook	com.facebook.Facebook	0	0	62732	25846	
2	2016-04-01 08:17:59	2016-04-01 08:48:04	com.facebook.Facebook	com.facebook.Facebook	2889588	79639	9942	5325	
3	2016-04-01 09:48:33	2016-04-01 10:19:04	com.facebook.Facebook	com.facebook.Facebook	3694	1184	0	0	
4	2016-04-01 10:50:05	2016-04-01 11:21:12	com.facebook.Facebook	com.facebook.Facebook	1739054	206785	0	0	
5	2016-04-01 15:33:48	2016-04-01 15:42:49	com.facebook.Facebook	com.facebook.Facebook	594344	31549	0	0	
6	2016-04-01 17:09:31	2016-04-01 17:36:21	com.facebook.Facebook	com.facebook.Facebook	108	68	0	0	
7	2016-04-01 17:41:27	2016-04-01 17:46:31	com.facebook.Facebook	com.facebook.Facebook	1486	1757	0	0	
8	2016-04-01 21:38:19	2016-04-01 21:52:28	com.facebook.Facebook	com.facebook.Facebook	65831	20659	0	0	
9	2016-04-02 08:17:13	2016-04-02 08:34:31	com.facebook.Facebook	com.facebook.Facebook	5613786	190301	0	0	
10	2016-04-02 08:54:55	2016-04-02 09:00:50	com.facebook.Facebook	com.facebook.Facebook	304959	6905	0	0	
11	2016-04-02 10:07:49	2016-04-02 10:29:07	com.facebook.Facebook	com.facebook.Facebook	191256	37647	0	0	
12	2016-04-02 12:00:12	2016-04-02 12:09:11	com.facebook.Facebook	com.facebook.Facebook	1330	1685	0	0	

# HEALTH



## Sources

## APPS



C25K



MyFitnessPal



As apps request permission to update your data, they will be added to the list.

## DEVICES



miPhone6



Sarah's Apple Watch



Dashboard



Health Data



Sources



Medical ID

# HEALTH SOURCES - DEVICES

## healthdb.sqlite

```
1 select datetime(creation_date+978307200,'unixepoch','localtime') as timestamp,  
2 name, manufacturer, model, hardware, software from source_devices  
3
```

	timestamp	ROWID	name	manufacturer	model	hardware	software
1	2015-09-28 21:18:52	1	__NONE__				
2	2015-09-28 21:19:43	2	iPhone	Apple	iPhone	iPhone7,2	9.0.1
3	2015-09-29 08:25:43	3	__NONE__				
4	2015-09-29 08:25:49	4	Apple Watch	Apple	Watch	Watch1,1	2.0
5	2015-10-02 12:00:24	5	iPhone	Apple	iPhone	iPhone7,2	9.0.2
6	2015-10-25 20:49:38	6	__NONE__				
7	2015-10-25 20:49:46	7	Apple Watch	Apple	Watch	Watch1,1	2.0

## Sources

## APPS



C25K



MyFitnessPal



As apps request permission to update your data, they will be added to the list.

## DEVICES



miPhone6



Sarah's Apple Watch



Dashboard



Health Data



Sources



Medical ID

# HEALTH SOURCES – APPS

## healthdb.sqlite

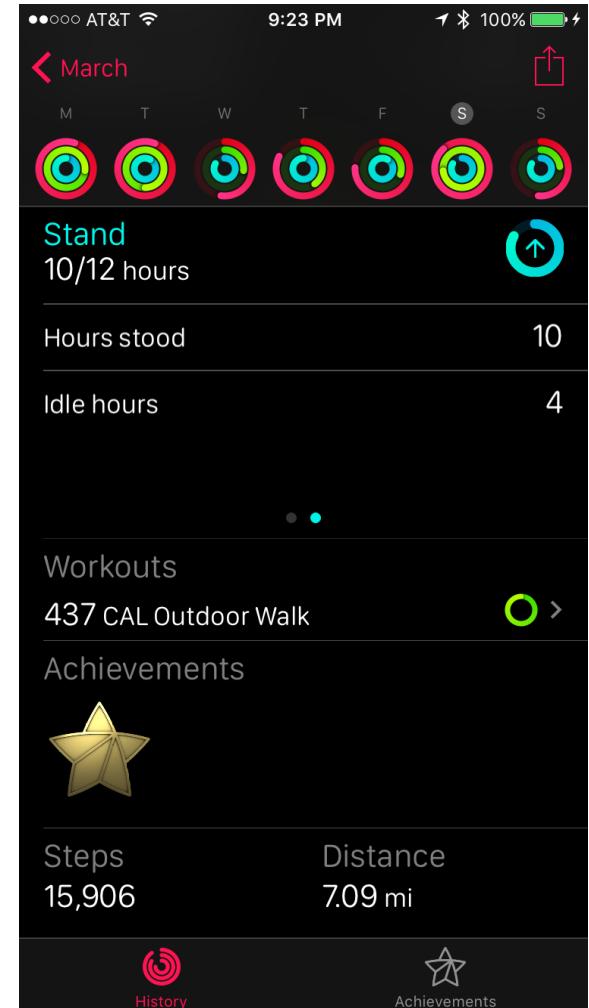
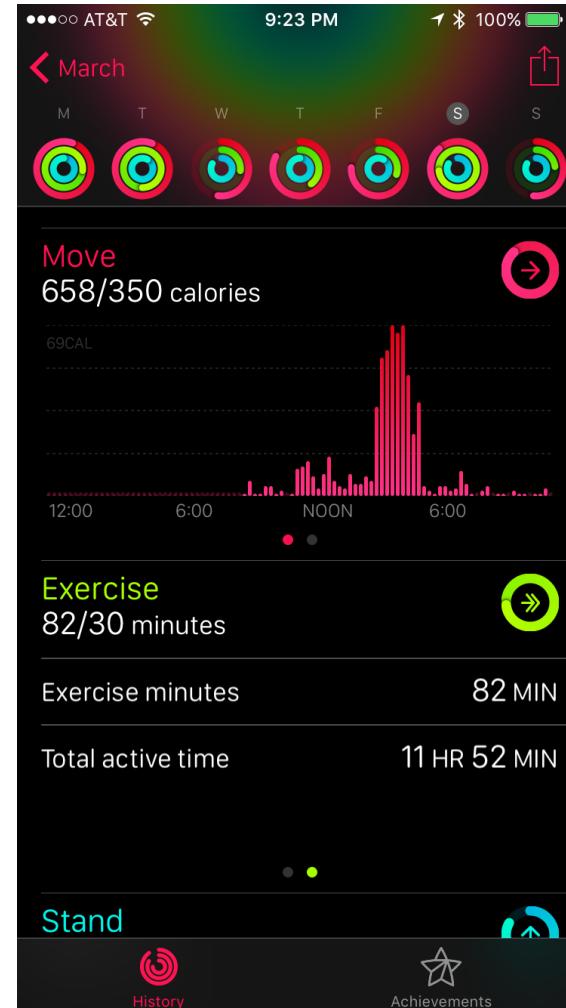
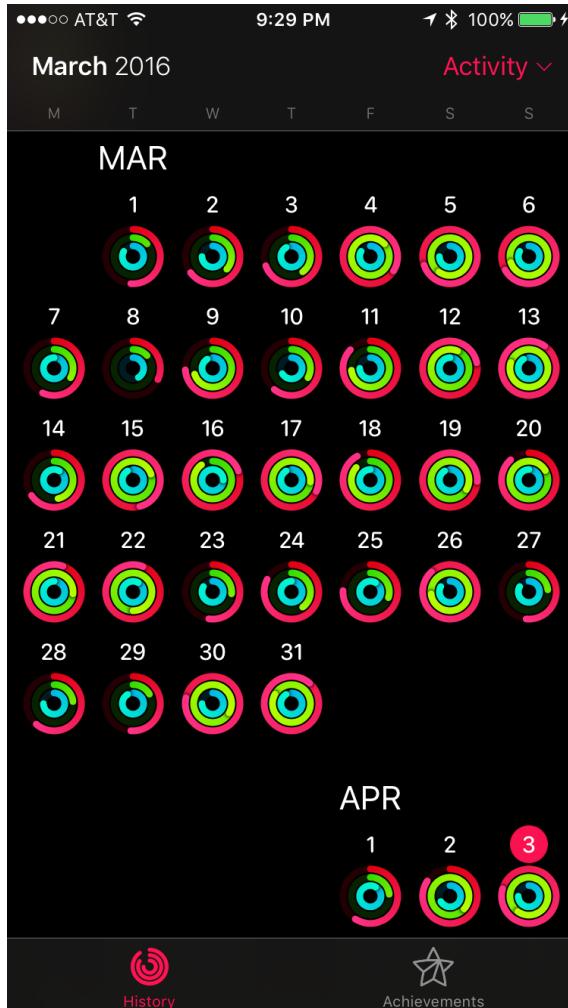
```
1 select name, bundle_id, source_options, local_device, product_type from sources
```

```
2
```

	name	bundle_id	source_options	local_device	product_type
1	miPhone6	com.apple.health.32BEA748-6D9B-...	2	1	iPhone7,2
2	Health	com.apple.Health	3	0	NULL
3	MyFitnessPal	com.myfitnesspal.mfp	5	0	NULL
4	C25K	com.zenlabs.c25k	5	0	NULL
5	Sarah's Apple Watch	com.apple.health.1A3E2A70-8921-...	2	0	Watch1,1

# ACTIVITY CACHE

## healthdb\_secure.sqlite



**Data Retention:**  
For All Time

**Timestamps:**  
Accurate

# ACTIVITY CACHE

## healthdb\_secure.sqlite

```
1 select datetime(energy_burned_goal_date+978307200,'unixepoch','utc') as "Energy Burned Goal Date",
2      datetime(cache_index+978307200,'unixepoch','utc') as "cache_index", energy_burned, energy_burned_goal,
3      active_hours, active_hours_goal, brisk_minutes, brisk_minutes_goal, steps, walk_distance
4 from activity_caches
```

	Energy Burned Goal Date	cache_index	energy_burned	energy_burned_goal	active_hours	active_hours_goal	brisk_minutes	brisk_minutes_goal	steps	walk_distance
329	2016-03-14 18:03:56	2016-03-14 04:00:00	229.079	350.0	13.0	12.0	14.0	30.0	4027.0	2881.04356095369
330	2016-03-14 18:03:56	2016-03-15 04:00:00	511.313	350.0	11.0	12.0	36.0	30.0	9962.0	7396.11327804928
331	2016-03-14 18:03:56	2016-03-16 04:00:00	418.704	350.0	15.0	12.0	27.0	30.0	6900.0	5157.84432498738
332	2016-03-14 18:03:56	2016-03-17 04:00:00	461.829	350.0	11.0	12.0	38.0	30.0	9701.0	7071.6955836229
333	2016-03-14 18:03:56	2016-03-18 04:00:00	322.419	350.0	12.0	12.0	26.0	30.0	6412.0	4798.06892250094
334	2016-03-14 18:03:56	2016-03-19 04:00:00	440.633	350.0	11.0	12.0	40.0	30.0	9643.0	7020.81993605674
335	2016-03-14 18:03:56	2016-03-20 04:00:00	309.39	350.0	12.0	12.0	35.0	30.0	5352.0	3965.97781493876
336	2016-03-21 19:23:36	2016-03-21 04:00:00	368.512	350.0	12.0	12.0	38.0	30.0	6647.0	5069.63263124716
337	2016-03-21 19:23:36	2016-03-22 04:00:00	371.162	350.0	13.0	12.0	45.0	30.0	7786.0	5912.25421512942
338	2016-03-21 19:23:36	2016-03-23 04:00:00	183.033	350.0	10.0	12.0	8.0	30.0	2366.0	1720.11841394124
339	2016-03-21 19:23:36	2016-03-24 04:00:00	289.15	350.0	12.0	12.0	12.0	30.0	4839.0	3453.24857764703
340	2016-03-21 19:23:36	2016-03-25 04:00:00	265.012	350.0	13.0	12.0	9.0	30.0	3635.0	2714.82200816309
341	2016-03-21 19:23:36	2016-03-26 04:00:00	658.617000000001	350.0	10.0	12.0	82.0	30.0	15906.0	11424.3456067285
342	2016-03-21 19:23:36	2016-03-27 04:00:00	180.65	350.0	10.0	12.0	7.0	30.0	2512.0	1876.62907064462

walk\_distance

2881.04356095369

7396.11327804928

5157.84432498738

7071.6955836229

4798.06892250094

7020.81993605674

3965.97781493876

5069.63263124716

5912.25421512942

1720.11841394124

3453.24857764703

2714.82200816309

11424.3456067285

# ACTIVITY CACHE

## healthdb\_secure.sqlite



11424.3456067285 meters in miles



≡ Examples    × Random

Input interpretation:

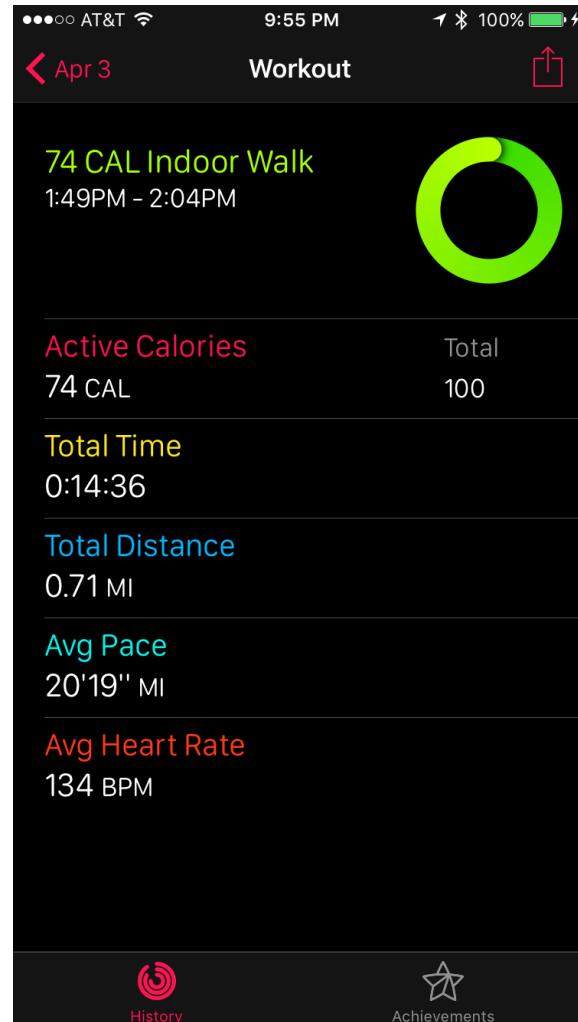
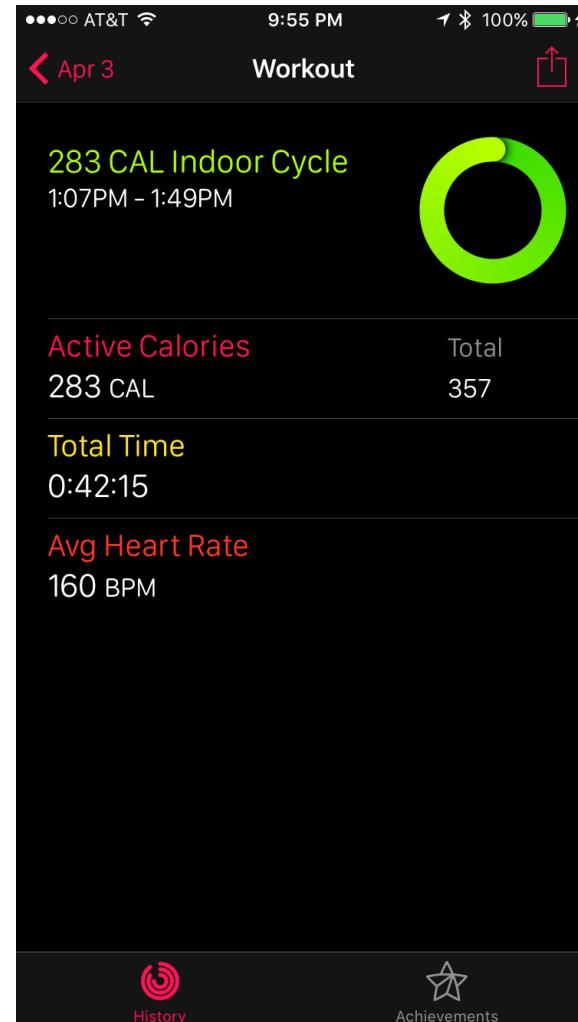
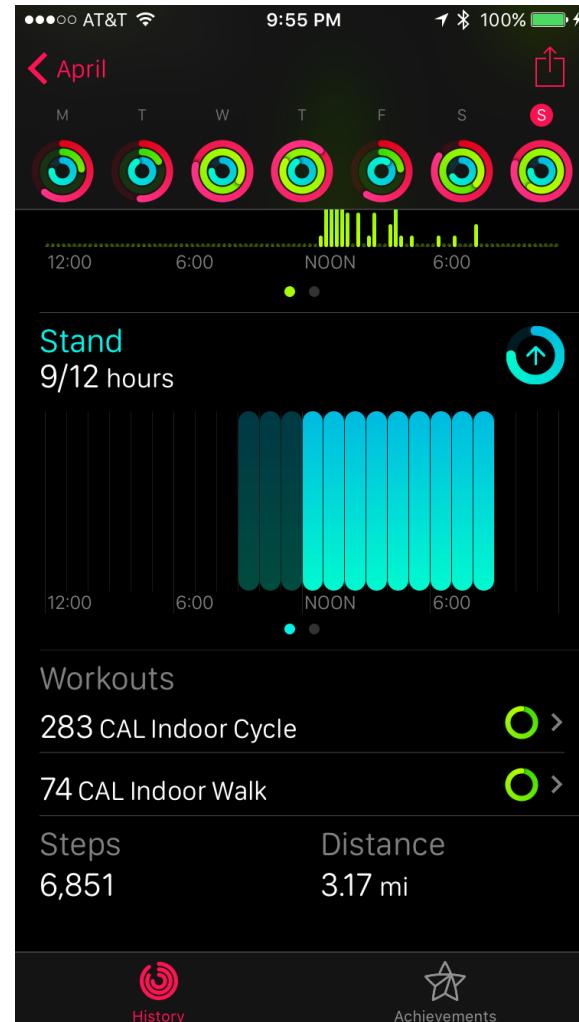
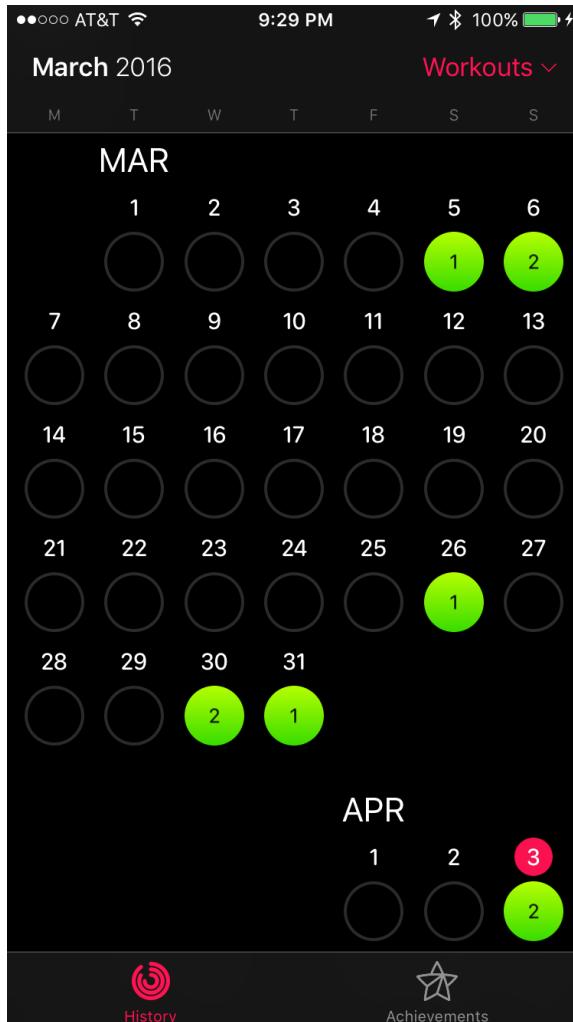
convert 11424.3456067285 meters to miles

Result:

7.0987592501842 miles

# WORKOUTS

## healthdb\_secure.sqlite



# WORKOUTS

## healthdb\_secure.sqlite

```

1 select samples.data_id, datetime(samples.start_date+978307200,'unixepoch','localtime') as "Start Date",
2      datetime(samples.end_date+978307200,'unixepoch','localtime') as "End Date", data_type, activity_type,
3      total_energy_burned, total_basal_energy_burned, total_distance, goal_type, goal from workouts
4      join samples on workouts.data_id = samples.data_id

```

	data_id	Start Date	End Date	data_type	activity_type	total_energy_burned	total_basal_energy_burned	total_distance	goal_type	goal
56	726977	2016-03-06 11:37:44	2016-03-06 12:11:23	79	13	178.624	58.37599999999997	0.301014298897014	0	NULL
57	773070	2016-03-26 15:40:28	2016-03-26 17:40:46	79	52	437.0819999999999	189.145999999995	8.94689508234034	0	NULL
58	782186	2016-03-30 07:17:00	2016-03-30 08:05:31	79	13	339.647	84.3370000000003	0.0	0	NULL
59	782187	2016-03-30 08:05:59	2016-03-30 08:17:33	79	37	74.531	19.932	0.888146075317753	0	NULL
60	783270	2016-03-31 07:15:54	2016-03-31 08:18:23	79	13	420.202000000001	108.378000000001	0.146758156347576	0	NULL
61	792862	2016-04-03 13:07:05	2016-04-03 13:49:21	79	13	283.935000000001	73.488	0.057244980888152	0	NULL
62	792866	2016-04-03 13:49:41	2016-04-03 14:04:18	79	52	74.8020000000001	25.10699999999999	1.15648563791718	0	NULL

**Workout Activity Types:**

13 = Indoor Cycle  
16 = Elliptical  
35 = Rower

37 = Indoor Run  
52 = Outdoor/Indoor Walk  
3000 = "Other"

**Goal Type:**  
2 = Time ("goal" in minutes)

# ACTIVITY “SAMPLES”

## healthdb\_secure.sqlite

```

1 select datetime(samples.start_date+978307200,'unixepoch','localtime') as "Start Date", datetime(samples.end_date+978307200,'unixepoch','localtime') as "End Date",
2 samples.data_id, data_type, quantity, original_quantity, unit_strings.unit_string, data_provenances.origin_device, data_provenances.origin_build,
3 data_provenances.local_device, data_provenances.local_build from samples
4 left outer join quantity_samples on samples.data_id = quantity_samples.data_id
5 left outer join unit_strings on quantity_samples.original_unit = unit_strings.RowID
6 left outer join objects on samples.data_id = objects.data_id
7 left outer join data_provenances on objects.provenance = data_provenances.RowID
8 where "Start Date" like '%2016-04-03%'

```

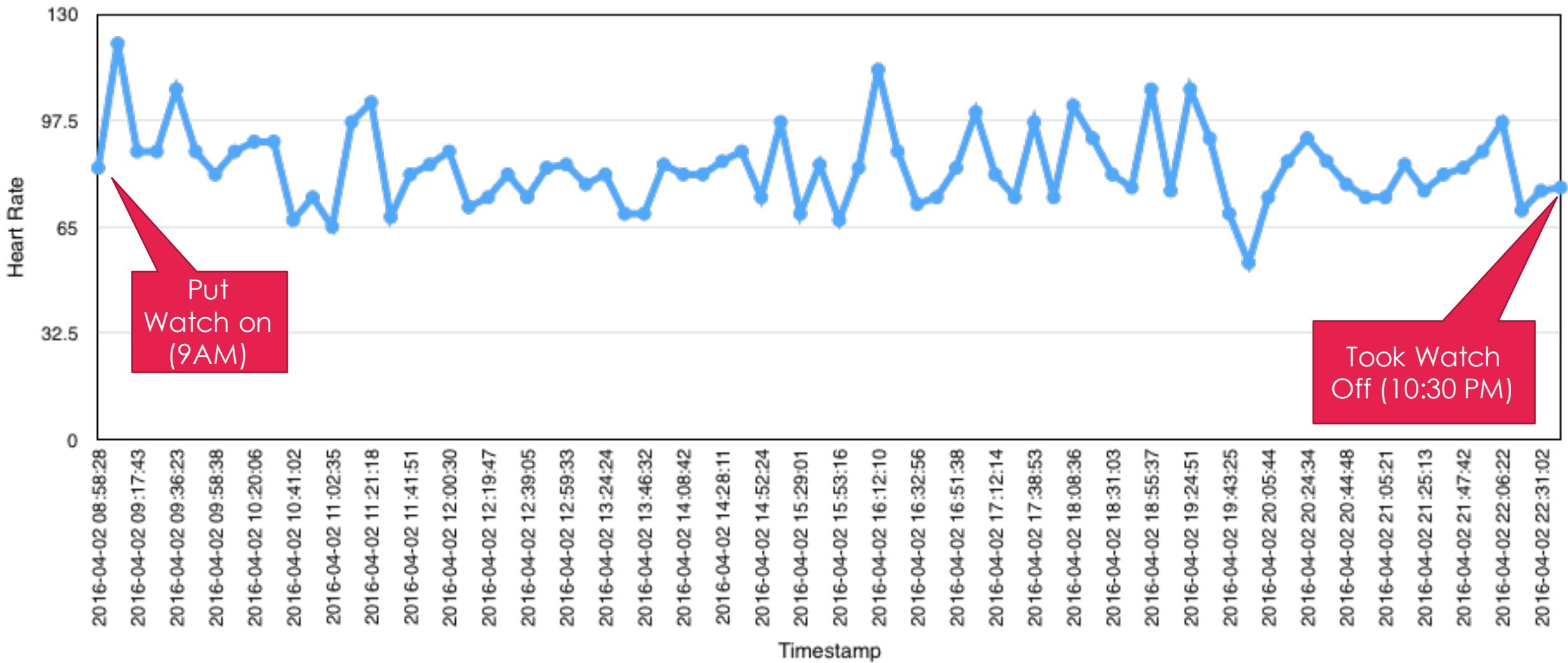
	Start Date	End Date	data_id	data_type	quantity	original_quantity	unit_string	origin_device	origin_build	local_device	local_build
4857	2016-04-03 16:40:00	2016-04-03 16:43:52	797742	8	19.0618415377103	NULL	NULL	Apple Watch	13S344	iPhone	13A452
4858	2016-04-03 16:40:00	2016-04-03 16:43:52	797745	7	28.0	NULL	NULL	Apple Watch	13S344	iPhone	13A452
4859	2016-04-03 16:40:38	2016-04-03 16:44:57	797509	7	115.0	NULL	NULL	iPhone	13A452	iPhone	13A452
4860	2016-04-03 16:40:38	2016-04-03 16:44:57	797516	8	80.3639120720327	NULL	NULL	iPhone	13A452	iPhone	13A452
4861	2016-04-03 16:41:20	2016-04-03 16:42:01	797739	10	0.5	500.0	cal	Apple Watch	13S344	iPhone	13A452
4862	2016-04-03 16:41:20	2016-04-03 16:42:01	797740	9	1.204	1204.0	cal	Apple Watch	13S344	iPhone	13A452
4863	2016-04-03 16:42:01	2016-04-03 16:43:03	797750	10	0.248	248.0	cal	Apple Watch	13S344	iPhone	13A452

**Data Types:**

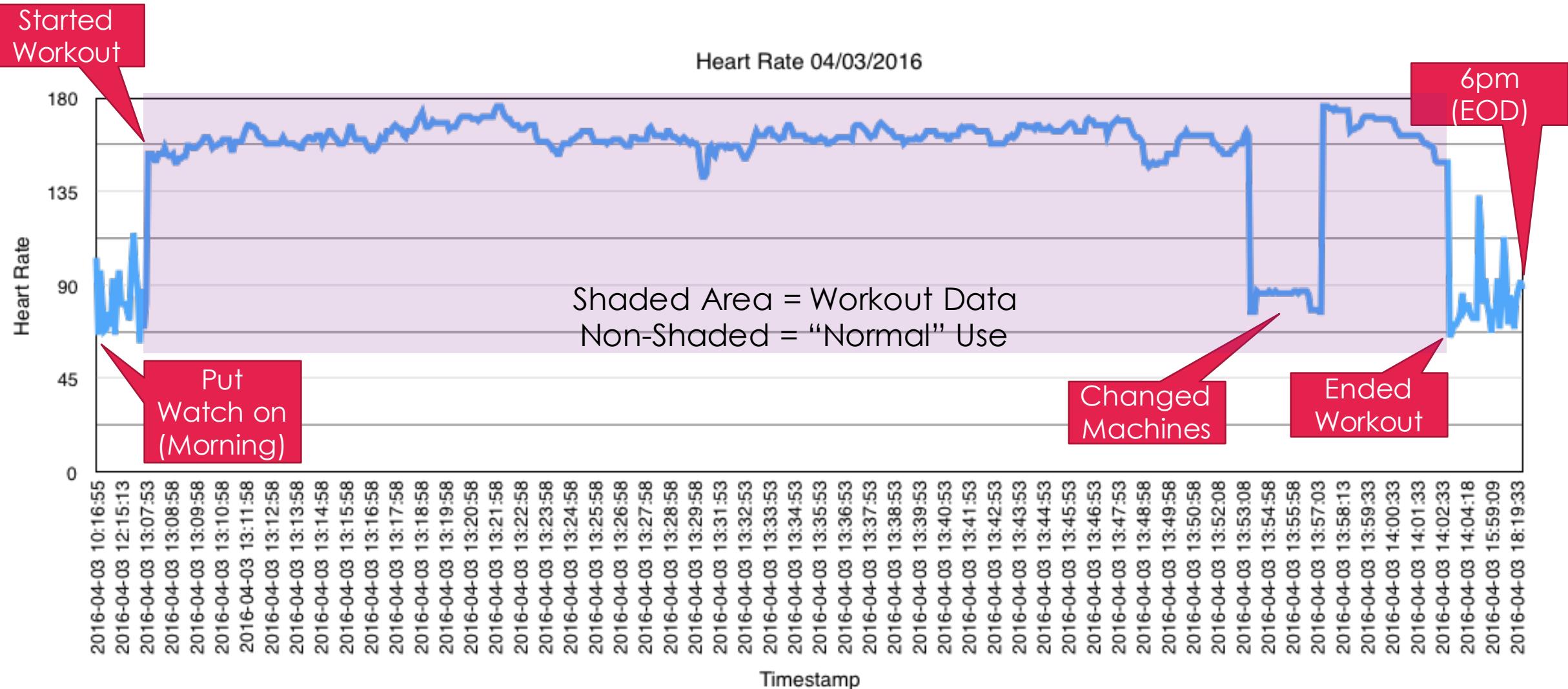
- 3 = Weight
- 5 = Heart Rate  
(original\_quantity)
- 7 = Steps
- 8 = Distance in Meters
- 9 = Resting Energy
- 10 = Active Energy
- 12 = Flights Climbed
- 20's ~ 30's = Nutrition
- 67 = Weekly Calorie Goal
- 70= Watch On
- 75 = Stand (Stood)
- 76 = Activity
- 79 = Workout
- 83 = Some Workouts

# HEART RATE/DAY (NON-WORKOUT) “NORMAL” SAMPLING

Heart Rate 04/02/2016

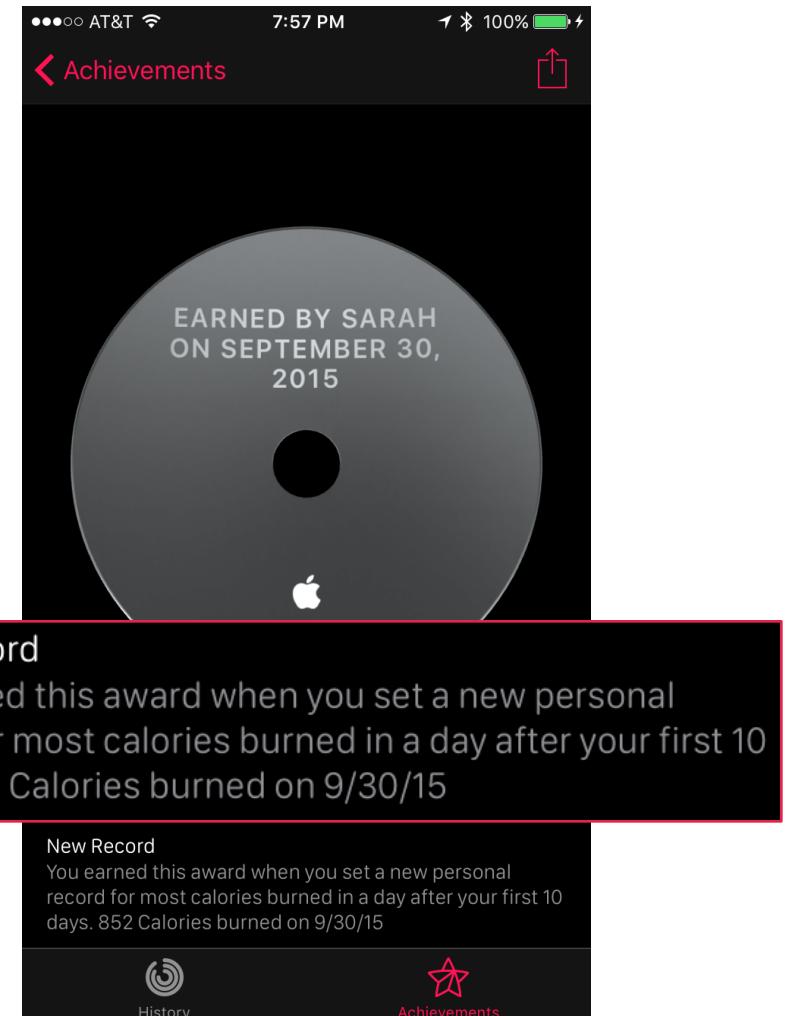
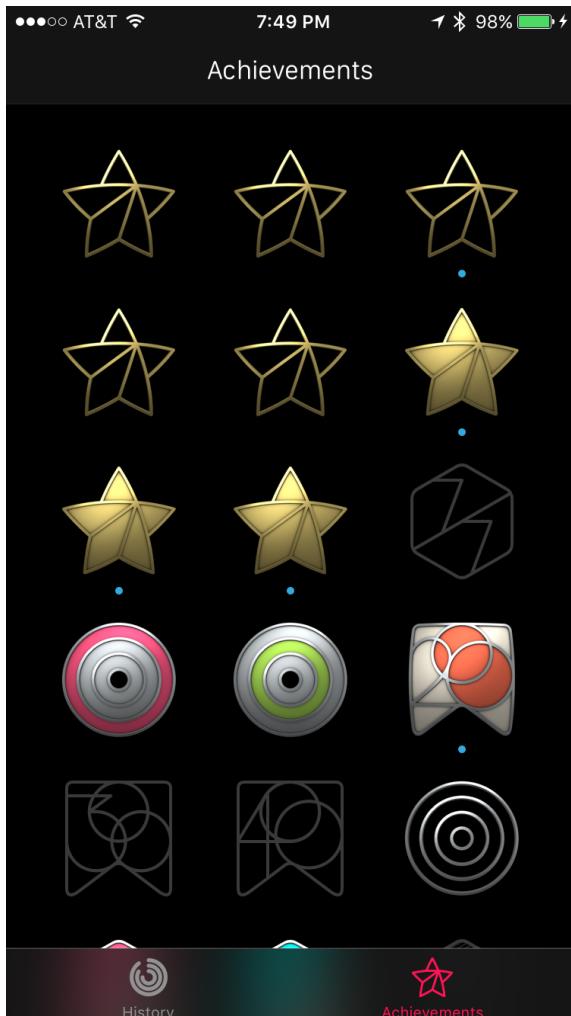


# HEART RATE/DAY (WORKOUT) HIGH SAMPLING



# ACHIEVEMENTS

## healthdb\_secure.sqlite



# ACHIEVEMENTS

## healthdb\_secure.sqlite

```
1 select datetime(completed_date+978307200,'unixepoch','localtime') as "Completed Timestamp",  
2 type, value, extra_data, viewed from achievements
```

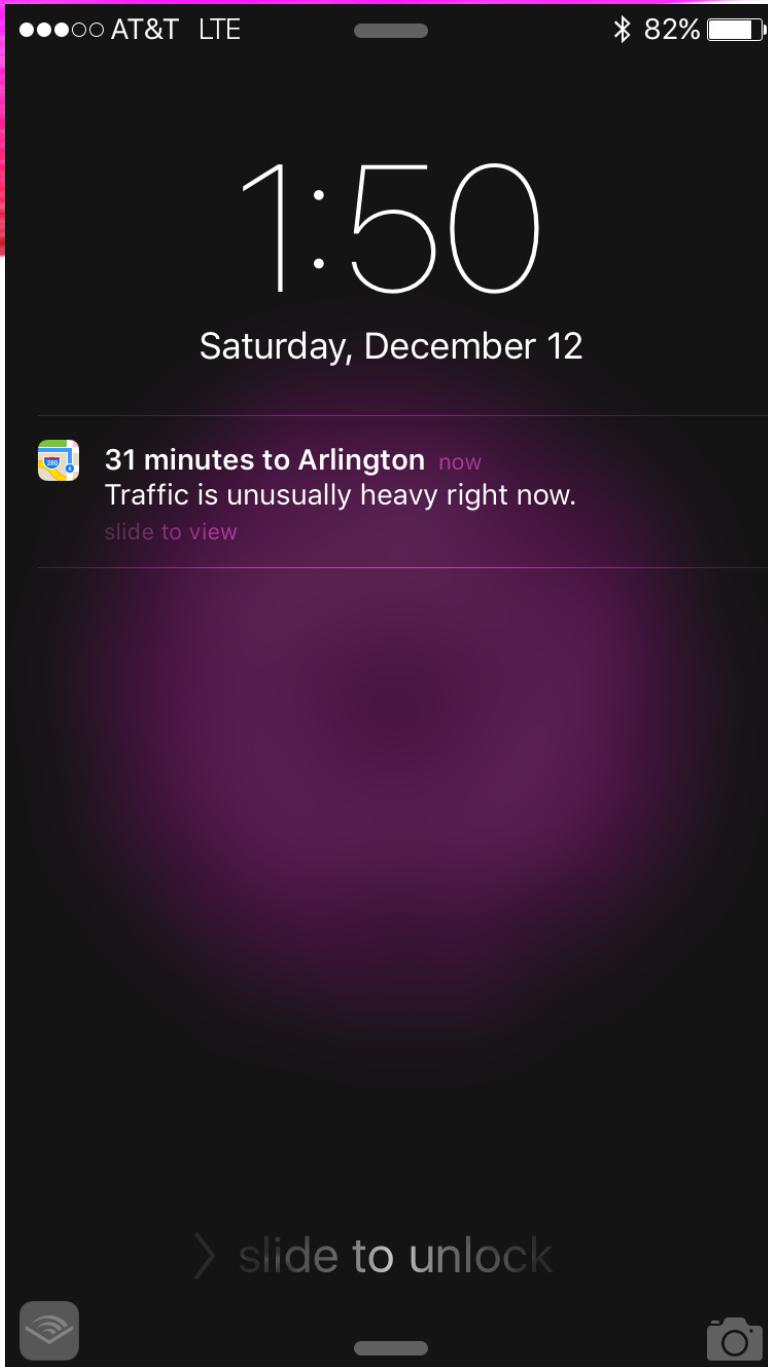
	Completed Timestamp	type	value	extra_data	viewed
29	2015-09-07 18:48:08	12	380.0	NULL	0
30	2015-09-14 23:46:06	12	400.0	NULL	1
31	2015-09-30 21:15:44	9	800.0	NULL	1
32	2015-09-30 00:00:00	6	852.04	NULL	1
33	2015-09-30 00:00:00	7	124.0	NULL	1
34	2015-10-18 20:30:52	17	100	NULL	1
35	2015-10-19 18:49:35	2	162.9230000000001	BLOB	0
36	2015-11-05 07:56:04	2	253.275	BLOB	0
37	2015-11-06 07:50:44	2	373.9710000000001	BLOB	1
38	2015-11-18 07:37:51	2	213.876	BLOB	0

### Achievement Type Examples:

- 1 = First <Exercise> Workout
- 2 = <Exercise> Workout Record
- 6 = New Record (Most Cals/Day)
- 7 = New Record (Most Exercise Min/Day)
- 9 = Move Goal 200%
- 12 = New Move Goal (Attained)
- 13 = Perfect Week (Move[1]/Stand[3]/Exercise[2?])
- 16 = Longest (Daily) Move Streak
- 17 = 100 Move Goal (Attained)

# LOCATION





# FREQUENT LOCATIONS (ROUTINED)

- User Viewable (View Yours!):
  - Settings ->
  - Privacy ->
  - Location Services ->
  - System Services ->
  - Frequent Locations
- Uses Location Services
- More like “Frequent” and “Recent Locations
- “Frequent” algorithm is unknown

# FREQUENT LOCATIONS (ROUTINED)

Settings -> Privacy -> Location Services -> System Services -> Frequent Locations

The image displays three screenshots of an iPhone's Settings app, illustrating the process of viewing frequent locations.

**Screenshot 1: System Services Frequent Locations**  
Shows the 'Frequent Locations' toggle switch turned on. Below it, a 'HISTORY' section lists recent locations:

- Arlington, Virginia (4 locations recorded 2/28/16 - 4/12/16)
- Washington, District of Columbia (1 location recorded on 4/9/16)
- Stafford, Virginia (2 locations recorded 3/28/16 - 4/8/16)
- Bay Lake, Florida (1 location first recorded on 3/16/16)
- California and Washington, United States (3 locations recorded 2/23/16 - 3/15/16)

**Screenshot 2: California and Washington**  
A map of the western United States focusing on California and Washington. Blue dots indicate recorded locations. Below the map, three specific locations are listed:

- S Disneyland Dr (5 visits recorded since February 23, 2016)
- S 176th St (4 visits recorded since March 12, 2016)
- Airport Way (3 visits recorded since February 28, 2016)

**Screenshot 3: Visits**  
A pie chart showing the distribution of visits across different locations. A small slice is labeled "Garden Court Bistro". Below the chart, a list of visit times and dates is provided:

- 3:30 AM - 9:00 AM  
2/27/16 - 2/28/16
- 7:00 PM - 9:00 PM  
2/25/16 - 2/26/16
- 2:00 PM - 10:45 AM  
2/24/16 - 2/25/16
- 11:45 AM - 2:00 PM  
2/24/16
- 12:45 AM - 11:45 AM  
2/23/16 - 2/24/16

# LOCATIONS (ROUTINED)

## cache\_encryptedB.db

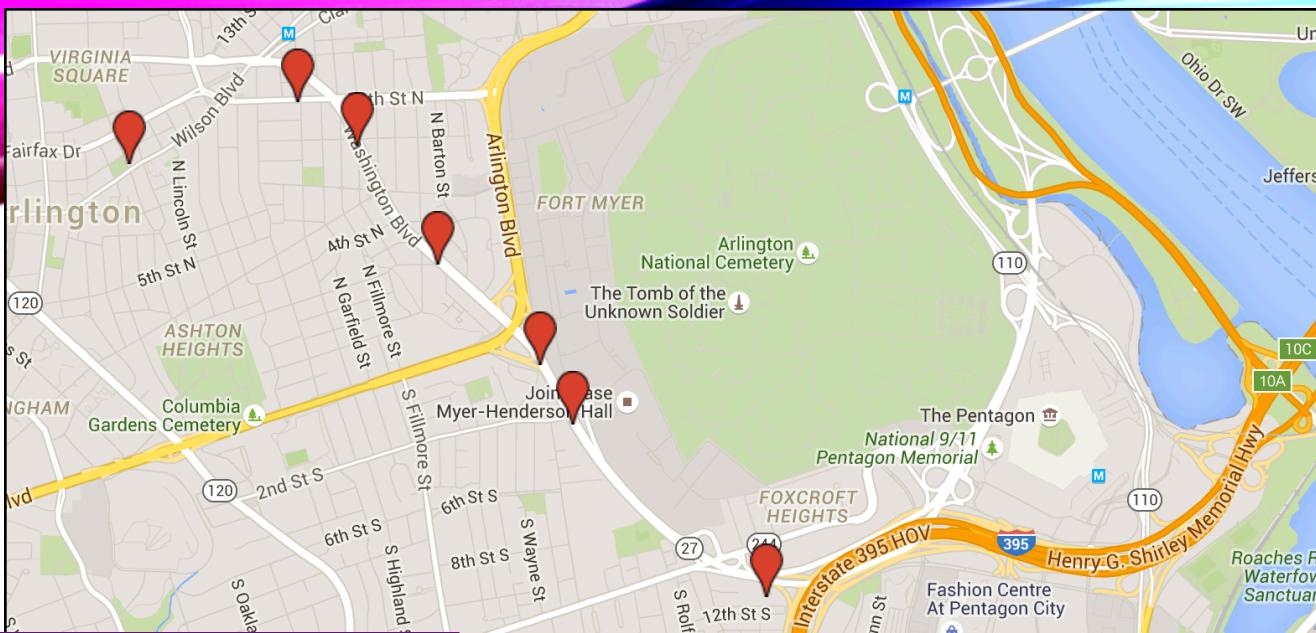
```
1 select datetime(timestamp+978307200,'unixepoch','localtime') as timestamp, Latitude, Longitude,  
2 Altitude, HorizontalAccuracy, VerticalAccuracy from location
```

	timestamp	Latitude	Longitude	Altitude	HorizontalAccuracy	VerticalAccuracy
1	2016-04-12 08:00:30	38.8819415801995	-77.1033633413063	83.9911193847656	65.0	49.8430881839981
2	2016-04-12 08:02:09	38.8844267330494	-77.0947589586276	89.2934923546982	5.0	3.6666666666666667
3	2016-04-12 08:03:07	38.8826809963573	-77.0916912951774	88.0339636439803	5.0	3.0
4	2016-04-12 08:04:08	38.8779316521612	-77.087568450791	72.6745093025534	5.0	3.0
5	2016-04-12 08:04:55	38.8739400334087	-77.0823709108653	61.1218382208071	5.0	3.0
6	2016-04-12 08:05:16	38.8715504400133	-77.0806616785783	53.1938431752319	5.0	3.0
7	2016-04-12 08:06:17	38.8646755004663	-77.0707583986941	47.0308895830595	5.0	3.0666666666666667
8	2016-04-12 08:07:17	38.8531212901097	-77.0745927480677	21.0405486171192	5.0	3.48387096774194
9	2016-04-12 08:08:18	38.8401141554299	-77.0860081836699	23.9693750537383	5.0	3.51612903225806
10	2016-04-12 08:09:18	38.8333078032593	-77.1033353773503	64.1204422513992	5.0	3.44827586206897

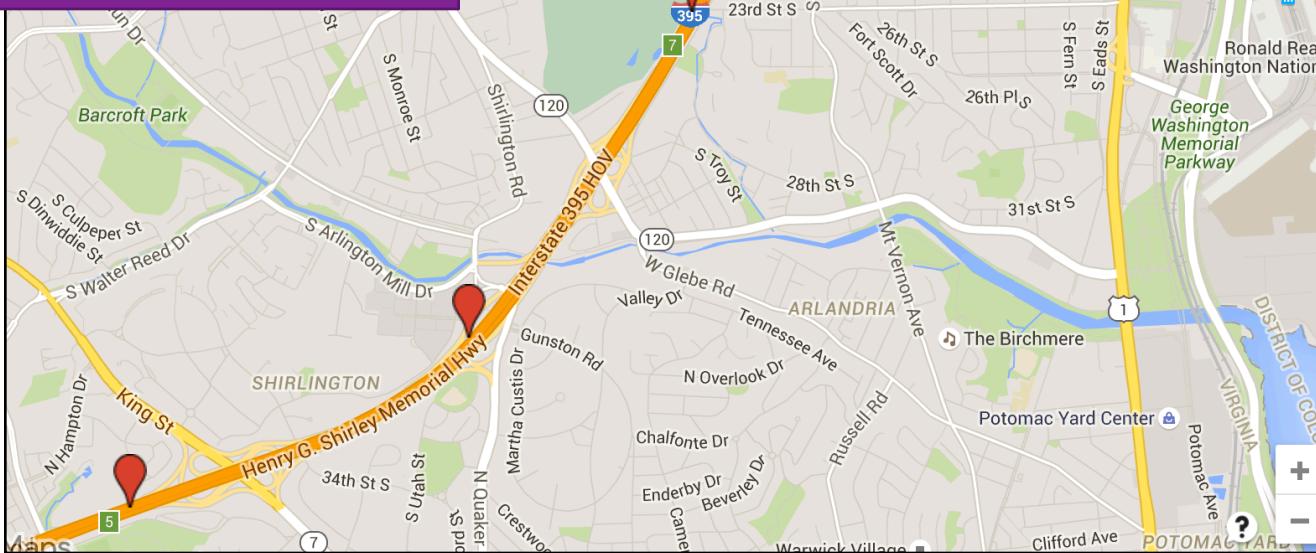
**Data Retention:**  
~1 Days

**Timestamps:**  
Accurate\*

**GPS Accuracy:**  
Close, but YMMV



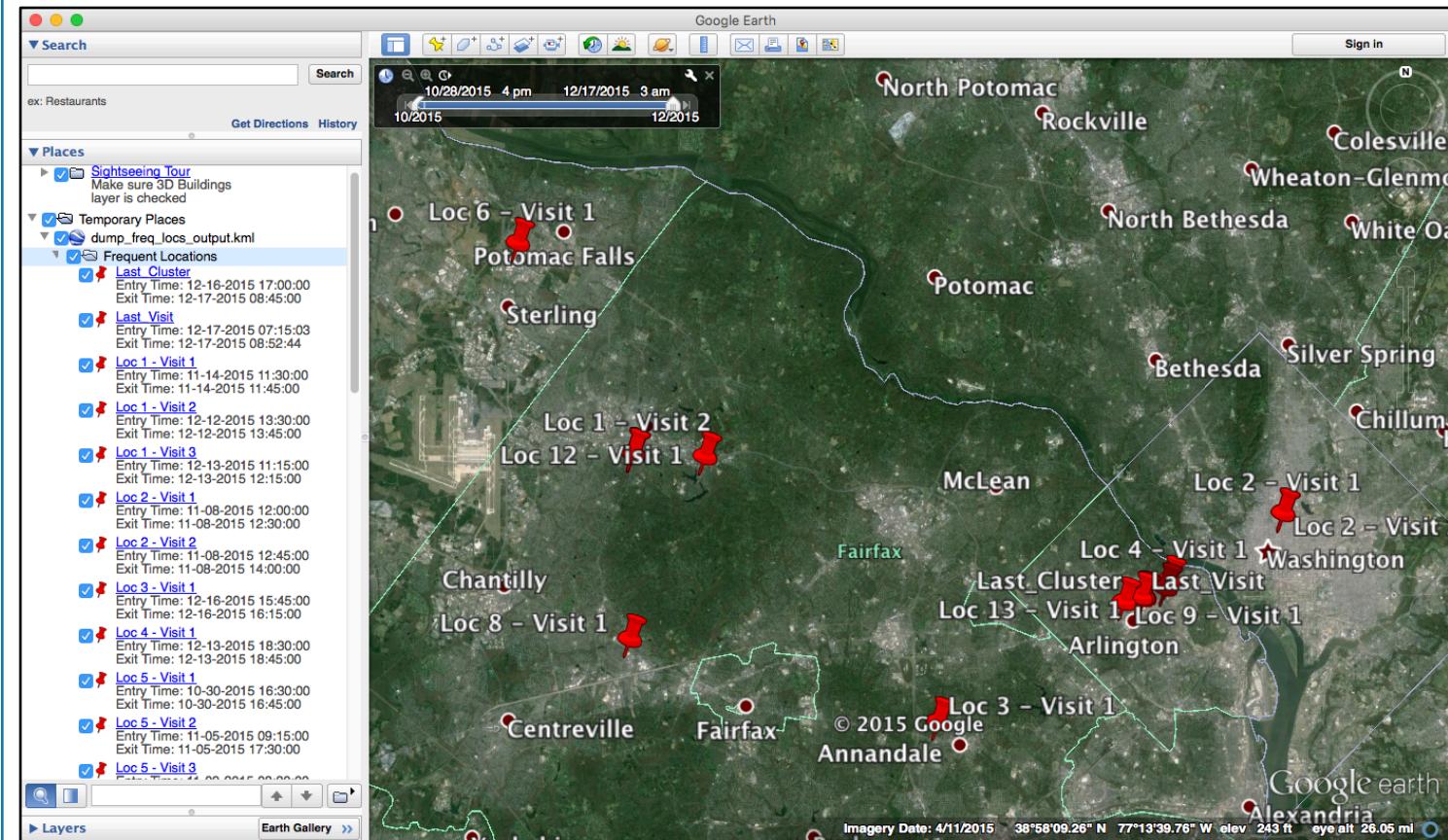
# Import Lat/Long into Google 'My Maps'



# FREQUENT LOCATIONS

## StateModel#.archive (1 or 2)

- Contains Historical Data
  - Not every location for all time
  - Uses algorithm to decide
- NSKeyedArchiver Property List Files
  - No immediate context to keys/values
  - Very manual analysis process. ☹
- Python Script!
  - <https://github.com/mac4n6/iOS-Frequent-Locations-Dumper>
- Outputs
  - KML
  - CSV
  - Textual (Example Next Slide)



# FREQUENT LOCATIONS

## StateModel#.archive (1 or 2)

```
#####
Location Entry Number 1:  
Entry Last Update Timestamp: 11-13-2015 02:08:19  
  
Location BLOB Contents:  
[DataPoints: 0]  
  
Hexdump Output:  
-----  
00000000: 0A A5 02 18 39 22 1A 32 33 30 31 E2 80 93 32 33 ....9".2301...23  
00000010: 37 37 20 43 6C 61 72 65 6E 64 6F 6E 20 42 6C 76 77 Clarendon Blv  
00000020: 64 32 EB 01 5A 1A 32 33 30 31 E2 80 93 32 33 37 d2..Z.2301...237  
00000030: 37 20 43 6C 61 72 65 6E 64 6F 6E 20 42 6C 76 64 7 Clarendon Blvd  
00000040: 5A 14 41 72 6C 69 6E 67 74 6F 6E 2C 20 56 41 20 Z.Arlington, VA  
00000050: 20 32 32 32 30 31 5A 0D 55 6E 69 74 65 64 20 53 22201Z.United S  
00000060: 74 61 74 65 73 7A A7 01 0A 0D 55 6E 69 74 65 64 tatesz....United  
00000070: 20 53 74 61 74 65 73 12 02 55 53 1A 08 56 69 72 States..US..Vir  
00000080: 67 69 6E 69 61 22 02 56 41 2A 09 41 72 6C 69 6E ginia".VA*.Arlin  
00000090: 67 74 6F 6E 32 09 41 72 6C 69 6E 67 74 6F 6E 3A gton2.Arlington:  
000000A0: 05 32 32 32 30 31 42 0A 43 6F 75 72 74 68 6F 75 .22201B.Courthou  
000000B0: 73 65 52 0E 43 6C 61 72 65 6E 64 6F 6E 20 42 6C seR.Clarendon Bl  
000000C0: 76 64 5A 0B 32 33 30 31 E2 80 93 32 33 37 37 62 vdZ.2301...2377b  
000000D0: 1A 32 33 30 31 E2 80 93 32 33 37 37 20 43 6C 61 .2301...2377 Cla  
000000E0: 72 65 6E 64 6F 6E 20 42 6C 76 64 8A 01 14 43 6C rendon Blvd...Cl  
000000F0: 61 72 65 6E 64 6F 6E 2D 43 6F 75 72 74 68 6F 75 arendon-Courthou  
00000100: 73 65 8A 01 0A 43 6F 75 72 74 68 6F 75 73 65 4A se...CourthouseJ  
00000110: 12 09 4A E9 3E A5 EF 71 43 40 11 31 6B 18 F4 94 ..J.>..qC@.1k...  
00000120: 45 53 C0 58 02 70 C2 3B ES.X.p.;  
None  
  
Hex Output:  
[0aa5021839221a32333031e280933233373720436c6172656e646f6e20426c766432eb015a1a323  
33031e280933233373720436c6172656e646f6e20426c76645a1441726c696e67746f6e2c2056412  
0203232320315a0d556e69746564205374617465737aa7010a0d556e69746564205374617465731  
20255531a0856697267696e6961220256412a0941726c696e67746f6e320941726c696e67746f6e3  
a05323232031420a436f757274686f757365520e436c6172656e646f6e20426c76645a0b3233303  
1e2809332333737621a32333031e280933233373720436c6172656e646f6e20426c76648a0114436  
c6172656e646f6e2d436f757274686f7573658a010a436f757274686f7573654a12094ae93ea5ef7  
1434011316b18f4944553c0580270c23b]
```

<b>Location Data:</b>	
Latitude:	38.8901452083
Longitude:	-77.0872245449
Confidence:	0.1
Uncertainty:	110.17950307
Update Timestamp:	11-13-2015 02:08:17
<b>Visits (Entry/Exits):</b>	
Visit Number:	1
Entry Timestamp:	11-11-2015 12:09:33
Exit Timestamp:	11-11-2015 13:08:03
<b>Transition Data:</b>	
Transition Number:	1
Start/Stop:	1
Motion Activity Type:	<>Not Populated><
Route UUID:	\$null
Start Timestamp:	11-11-2015 13:03:03
Stop Timestamp:	11-21-2015 13:13:15

# CELL LOCATIONS

## cache\_encryptedA.db

## lockCache\_encryptedA.db

```
1 select datetime(timestamp+978307200,'unixepoch','localtime') as Timestamp,  
2 latitude, longitude, mcc, mnc, tac, ci, uarfcn  
3 from LteCellLocation
```

	Timestamp	Latitude	Longitude	MCC	MNC	TAC	CI	UARFCN
781	2016-04-09 16:35:05	38.89591594	-77.02227692	310	120	6152	51742266	-1
782	2016-04-09 16:35:05	38.90686029	-77.04580937	310	260	20234	10298625	-1
783	2016-04-09 16:41:23	38.88934702	-77.03765539	310	410	4638	168769552	-1
784	2016-04-09 16:41:23	38.8708062	-77.00907997	310	410	4631	167985533	-1
785	2016-04-09 16:41:23	38.88175763	-77.03926338	310	410	4638	168769546	-1
786	2016-04-09 16:41:23	38.87029243	-76.99438353	311	480	27400	104194848	-1
787	2016-04-09 16:41:23	38.86216224	-77.06727286	311	870	44929	82579467	-1
788	2016-04-09 16:41:23	38.88934702	-77.01273611	311	480	27410	104319008	-1

**Data Retention:**  
~1 Week  
(Varies Per Table)

**Timestamps:**  
Accurate\*

**GPS Accuracy:**  
Within General Area  
(See Next Slide)

**Many Other Tables:**

- CDMA
- SCDMA
- LTE
- WIFI
- “Indoor”
- Application/“WTW”

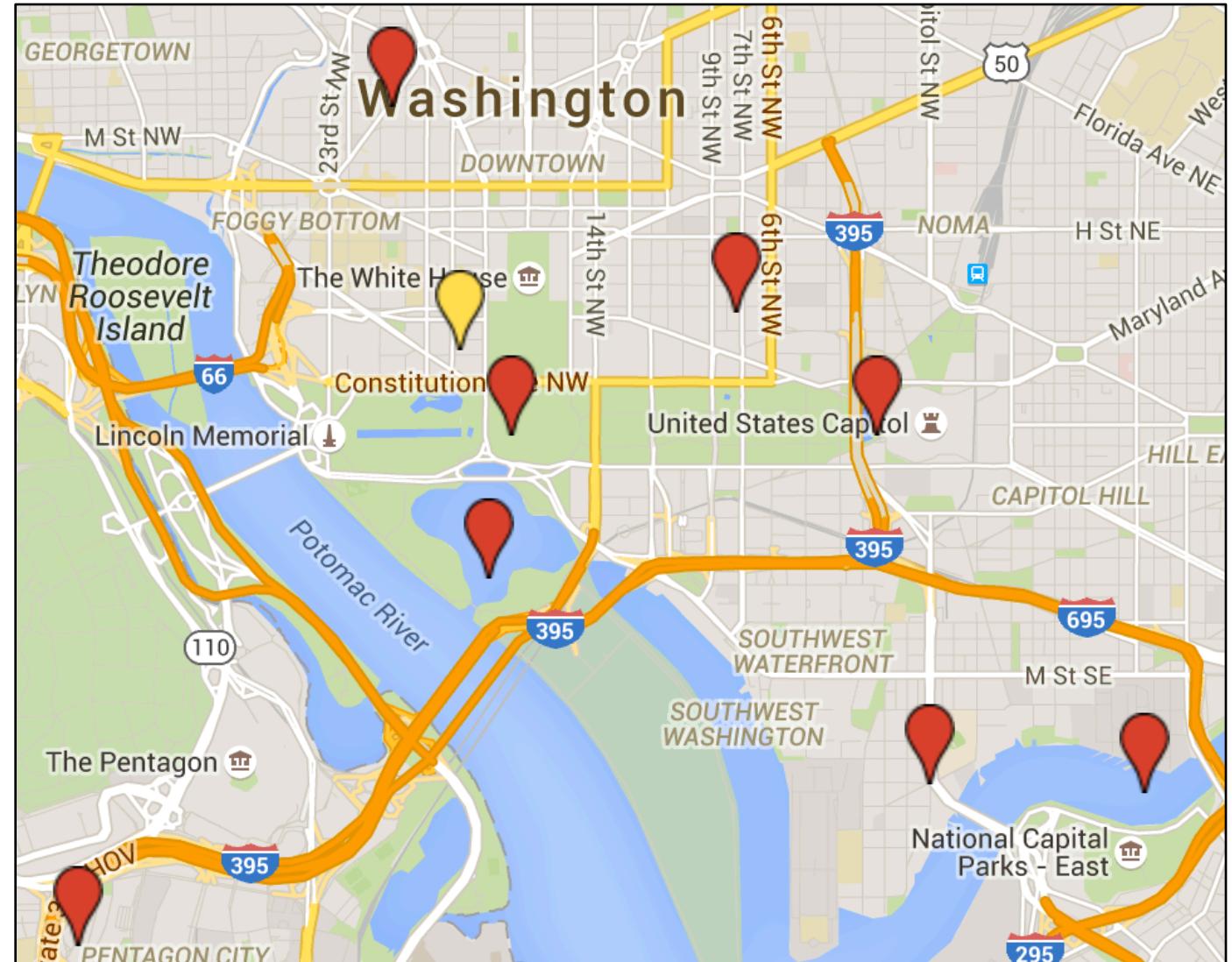
# CELL LOCATIONS

## cache\_encryptedA.db

**Yellow Point:**  
My Actual Location

**Red Points:**  
“LTE Cell Locations”

**Warning:**  
Locations are in  
general area, NOT  
exact area



# WIFI LOCATIONS

## cache\_encryptedB.db

```
1 select datetime(timestamp+978307200,'unixepoch','localtime') as Timestamp,  
2 MAC, Channel, Latitude, Longitude  
3 from WifiLocation  
4 order by Timestamp
```

	Timestamp	MAC	Channel	Latitude	Longitude
283	2016-04-09 14:06:26	202552290261808	1	38.89331141	-77.03996022
284	2016-04-09 14:06:26	202552290420464	1	38.89339663	-77.04031012
285	2016-04-09 14:06:26	202552290802784	11	38.89340956	-77.04041513
286	2016-04-09 14:06:26	202552290803296	11	38.89345488	-77.04029103
287	2016-04-09 14:06:26	202552290803904	6	38.89339626	-77.04028187
288	2016-04-09 14:06:26	202552291496976	1	38.89326488	-77.04264756
289	2016-04-09 14:06:26	202552291958080	11	38.8921614	-77.04037459

**Data Retention:**  
~4 Days

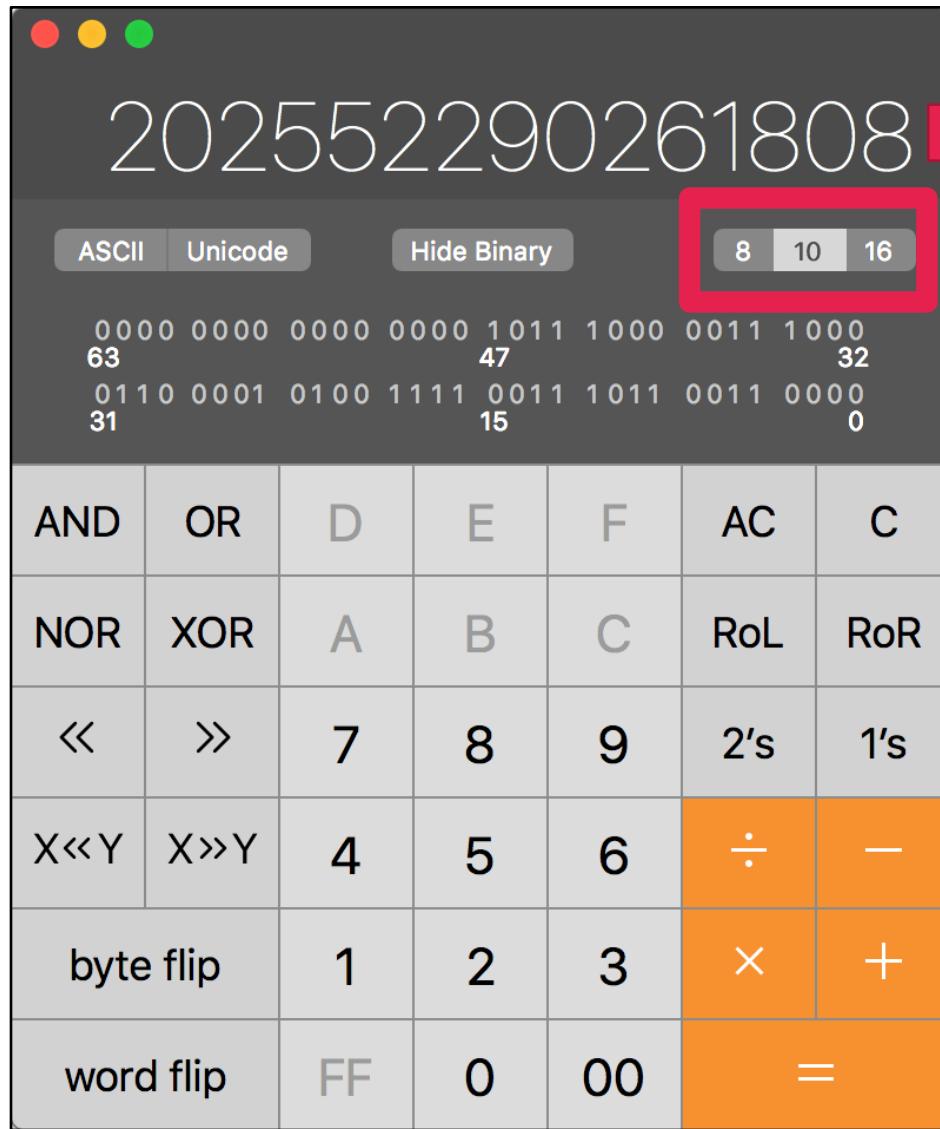
**Timestamps:**  
Accurate\*

**GPS Accuracy:**  
Within General Area

**MAC Address:**  
Stored in Base10

# WIFI LOCATIONS

## cache\_encryptedB.db



202552290261808

ASCII Unicode Hide Binary 8 10 16

0000	0000	0000	0000	1011	1000	0011	1000
63				47			32
0110	0001	0100	1111	0011	1011	0011	0000
31				15			0

AND OR D E F AC C

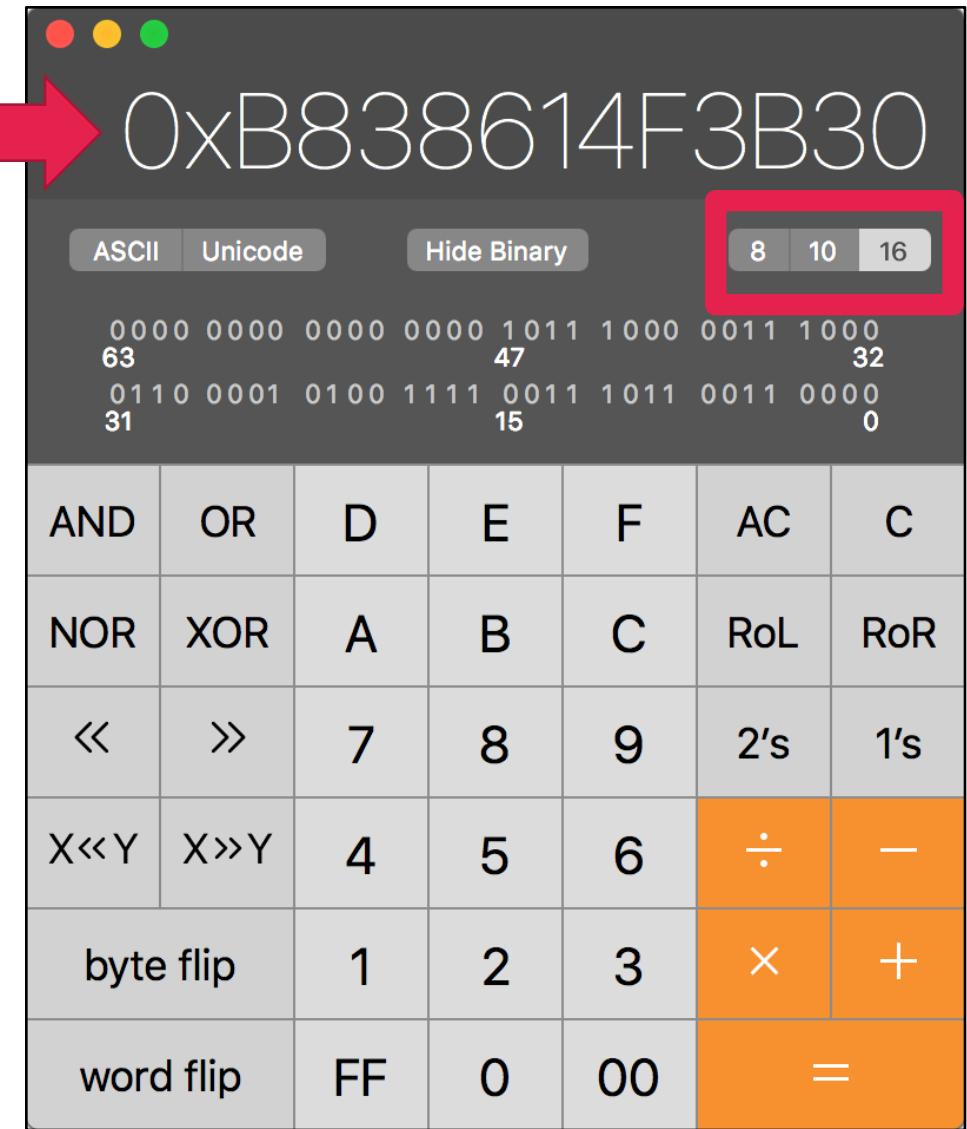
NOR XOR A B C RoL RoR

« » 7 8 9 2's 1's

X«Y X»Y 4 5 6 ÷ -

byte flip 1 2 3 × +

word flip FF 0 00 =



0xB838614F3B30

ASCII Unicode Hide Binary 8 10 16

0000	0000	0000	0000	1011	1000	0011	1000
63				47			32
0110	0001	0100	1111	0011	1011	0011	0000
31				15			0

AND OR D E F AC C

NOR XOR A B C RoL RoR

« » 7 8 9 2's 1's

X«Y X»Y 4 5 6 ÷ -

byte flip 1 2 3 × +

word flip FF 0 00 =

https://wigle.net/search

View Uploads Info Stats Tools

## Network Search

General Search Network Detail

### Query for networks

Latitude: 47.25264 to: 47.25265 Longitude: -87.256243 to: -87.256244

Search Radius Tolerance(+/- degrees): 0.010

BSSID/MAC: b8:38:61:4f:3b:30

SSID / Network Name (exact match): foobar

SSID / Network Name (wildcards<sup>1</sup>: % and \_: foobar%

Last Observed: 2001092517454

Must Be a FreeNet  Must Be a Commercial Pay Net  Only Networks I Was the First to Discover

Addresses are for the U.S. only (2002 Census data)

Street Address: 1600 Pennsylvania Ave State: DC Zip: 20502

**Query** **Reset**

<sup>1</sup> SSID cannot start with a wildcard. '%' means zero-or-more characters, '\_' means a single character.

<< showing records 1 to 1 >>

Map	Net ID	SSID	Name	Type	First Seen	Most Recently	Crypto	Est. Lat	Est. Long	Channel	Bcn Int.	QoS	Found by Me	Free	Pay	Comment
map	b8:38:61:4F:3B:30	OAS_OPEN	infra		2014-10-05 15:51:05	2015-05-09 14:45:51	?	38.89290619	-77.03944397	1			2			<b>add comment</b>

### Network Location

Click for interactive map

# LOCATIONS SCRAPER PYTHON SCRIPT

- Gimme All the Locations!
  - Routined
  - Locationd
- Python Script!
  - Will Release Soon!
  - Making code less redundant
  - Need it NOW?  
Just ask, politely.
- Outputs
  - KML
  - CSV

