518.5
Advanced Analysis Topics

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

518.5
Advanced Analysis Topics

SANS

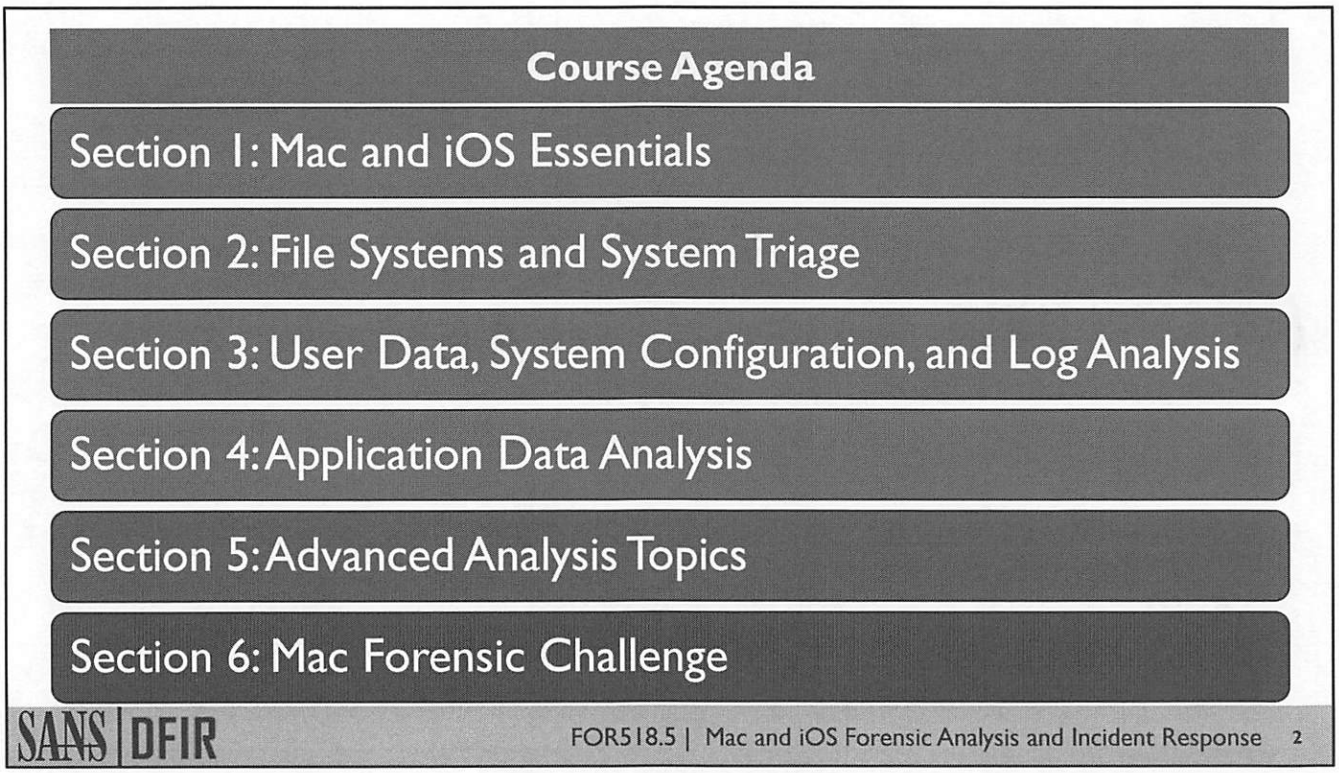
THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response

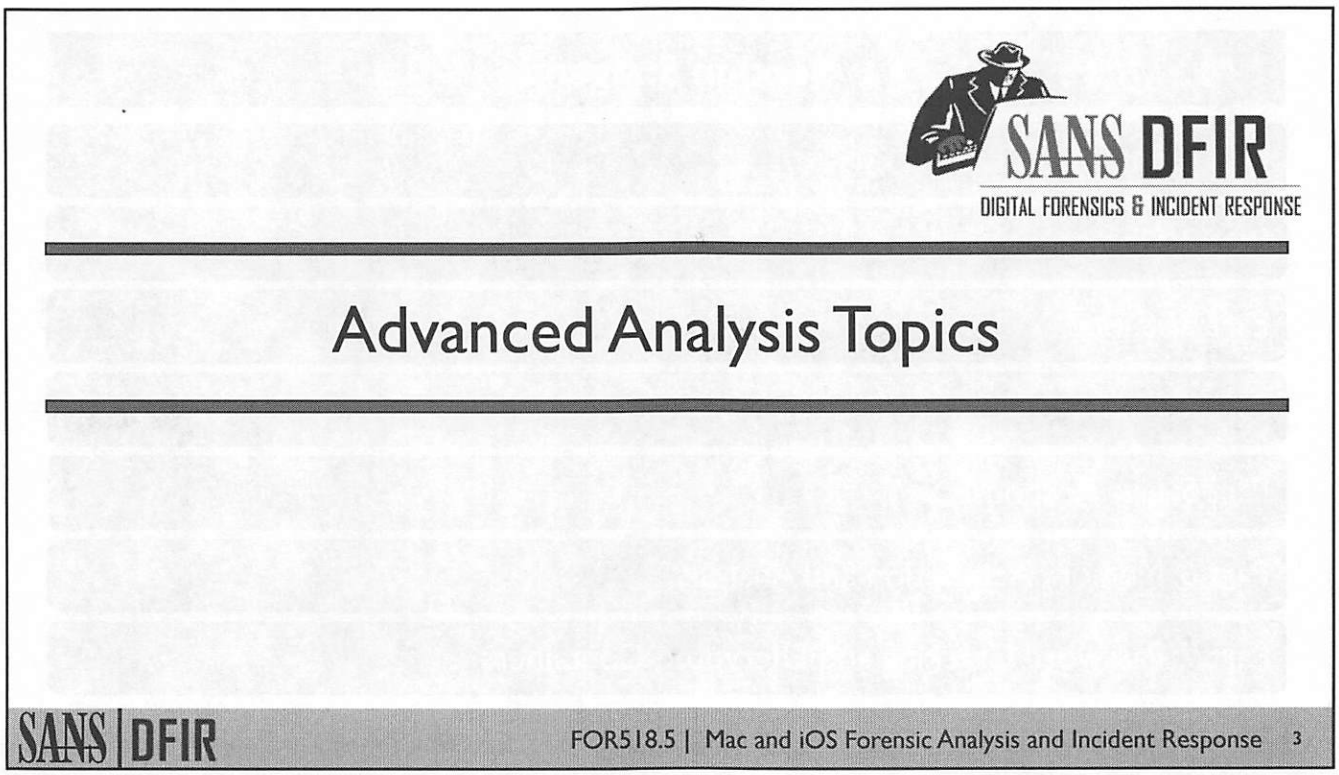
FOR518 Section 5: Advanced Analysis Topics

Author: Sarah Edwards
oompa@csh.rit.edu
mac4n6.com
http://twitter.com/iamevltwin

https://digital-forensics.sans.org/
http://twitter.com/sansforensics



This page intentionally left blank.

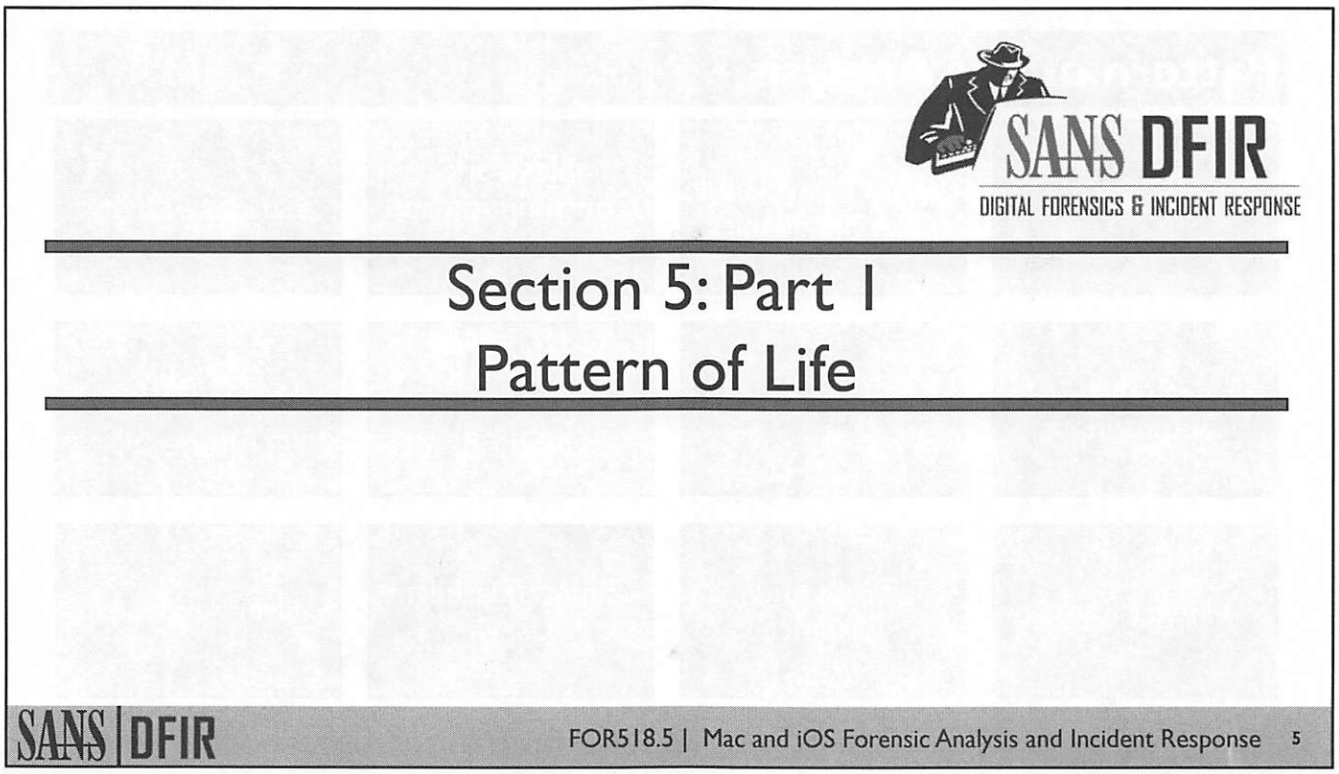


This page intentionally left blank.

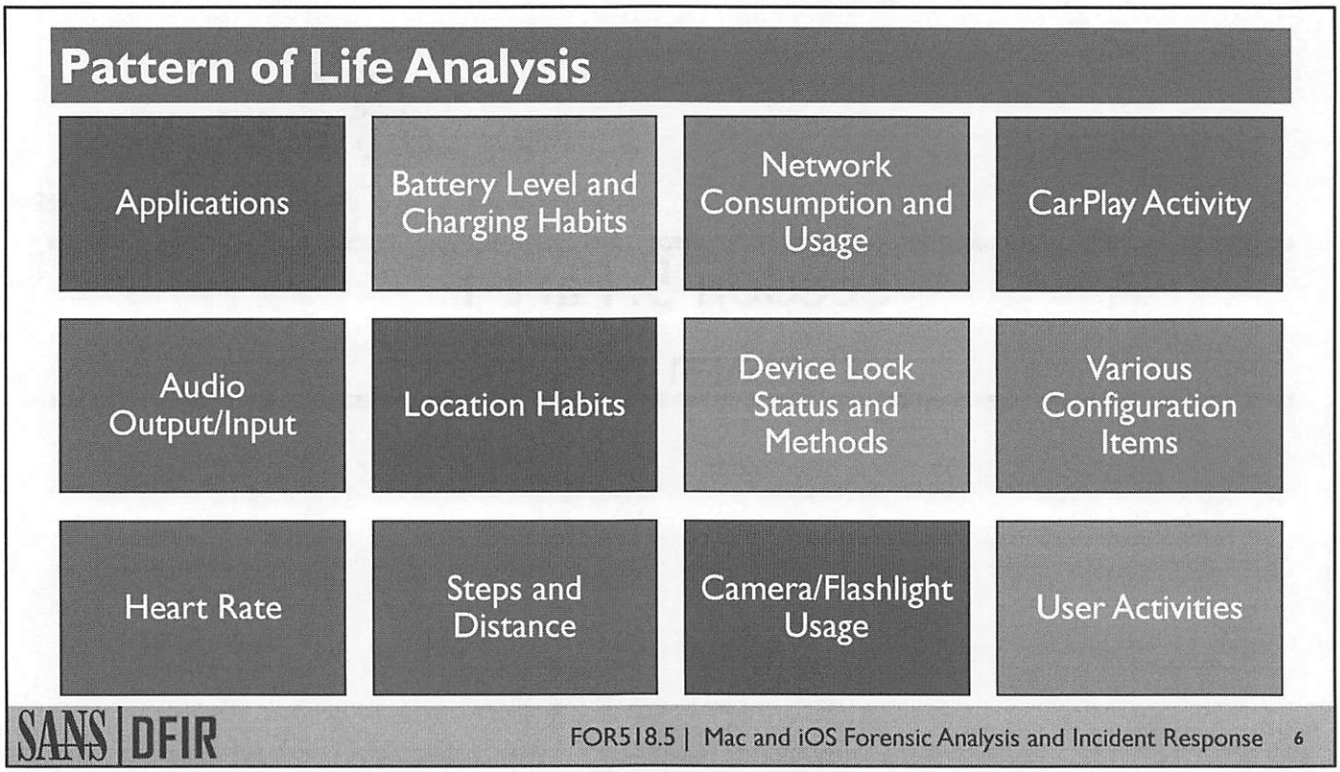
Section 5: Agenda

- Part 1: Pattern of Life
- Part 2: Document Versions
- Part 3: iCloud
- Part 4: Malware and Intrusion Analysis
- Part 5: Live Response
- Part 6: Memory Acquisition and Analysis
- Part 7: Password Cracking and Encrypted Containers

This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.

Application Usage – Apps Used on macOS and iOS (Physical Only) knowledgeC.db [~4 Weeks]

```
SELECT
datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH') as "ENTRY CREATION",
ZOBJECT.ZVALUESTRING AS "BUNDLE ID",
CASE ZOBJECT.ZSTARTDAYOFWEEK
    WHEN "1" THEN "Sunday"
    WHEN "2" THEN "Monday"
    WHEN "3" THEN "Tuesday"
    WHEN "4" THEN "Wednesday"
    WHEN "5" THEN "Thursday"
    WHEN "6" THEN "Friday"
    WHEN "7" THEN "Saturday"
END "DAY OF WEEK",
ZOBJECT.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",
datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH') as "START",
datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH') as "END",
(ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS"
FROM ZOBJECT
```

	ENTRY CREATION	BUNDLE ID	DAY OF WEEK	GMT OFFSET	START	END	USAGE IN SECONDS
211	2018-08-21 21:59:33	com.contextoptional.OpenTable	Tuesday	-4	2018-08-21 21:58:04	2018-08-21 21:59:33	89
212	2018-08-21 21:59:50	com.apple.MobileSMS	Tuesday	-4	2018-08-21 21:59:35	2018-08-21 21:59:49	14
213	2018-08-21 22:02:03	com.atebits.Tweetie2	Tuesday	-4	2018-08-21 21:59:50	2018-08-21 22:02:03	133
214	2018-08-21 22:03:23	com.contextoptional.OpenTable	Tuesday	-4	2018-08-21 22:02:05	2018-08-21 22:03:23	78
215	2018-08-21 22:05:09	com.apple.Fitness.activity-widget	Tuesday	-4	2018-08-21 22:05:07	2018-08-21 22:05:09	2
216	2018-08-21 22:05:09	com.waze.iphone.today	Tuesday	-4	2018-08-21 22:05:07	2018-08-21 22:05:09	2
217	2018-08-21 22:05:10	com.wunderground.weatherunderground.weatherwidget	Tuesday	-4	2018-08-21 22:05:07	2018-08-21 22:05:10	3
218	2018-08-21 22:05:11	com.apple.news.widget	Tuesday	-4	2018-08-21 22:05:07	2018-08-21 22:05:11	4
219	2018-08-21 22:11:33	com.atebits.Tweetie2	Tuesday	-4	2018-08-21 22:03:24	2018-08-21 22:11:33	489
220	2018-08-21 22:16:45	com.atebits.Tweetie2	Tuesday	-4	2018-08-21 22:13:13	2018-08-21 22:16:45	212

SANS | DFIR FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 7

macOS: `~/Library/Application Support/Knowledge/knowledgeC.db`
iOS: `/private/var/mobile/Library/CoreDuet/Knowledge/knowledgeC.db`

The `knowledgeC.db` keeps track of many things, one of the most useful items is application usage – start and end times for each application that is used. The app is stored using the bundle ID for the application.

On macOS, a very similar artifact is CoreAnalytics data files. More information (and a script!) can be found here: `https://www.crowdstrike.com/blog/i-know-what-you-did-last-month-a-new-artifact-of-execution-on-macos-10-13/`

References:
`https://www.mac4n6.com/blog/2018/8/5/knowledge-is-power-using-the-knowledgecdb-database-on-macos-and-ios-to-determine-precise-user-and-application-usage`
`https://www.mac4n6.com/blog/2018/9/12/knowledge-is-power-ii-a-day-in-the-life-of-my-iphone-using-knowledgecdb`
`https://www.mac4n6.com/blog/2018/12/16/on-the-third-day-of-apollo-my-true-love-gave-to-me-application-usage-to-determine-who-has-been-naughty-or-nice`
`https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series`
`https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie`

Application Usage – Apps Used (iOS, Physical, or sysdiagnose) CurrentPowerlog.PLSQL [~3 days] and /Archives

	ADJUSTED_TIMESTAMP	BUNDLE_ID	APPROLE	DISPLAY	LEVEL	ORIENTATION	SCREENWEIGHT	ORIGINAL_SCREEN_STATE_TIMESTAMP	OFFSET_TIMESTAMP	TIME_OFFSET
846	2018-09-17 19:30:52	com.apple.lock-screen	3	0	1050.0	1	1.0	2018-09-17 19:30:39	2018-09-17 16:02:01	13.1328829526901
847	2018-09-17 19:30:52	com.apple.springboard.home-screen	1	3	0.0	NULL	1.0	2018-09-17 19:30:39	2018-09-17 16:02:01	13.1328829526901
848	2018-09-17 19:30:52	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:30:39	2018-09-17 16:02:01	13.1328829526901
849	2018-09-17 19:31:03	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:30:50	2018-09-17 16:02:01	13.1328829526901
850	2018-09-17 19:31:03	com.apple.carplay.oem	3	3	8.0	NULL	1.0	2018-09-17 19:30:50	2018-09-17 16:02:01	13.1328829526901
851	2018-09-17 19:31:03	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:30:50	2018-09-17 16:02:01	13.1328829526901
852	2018-09-17 19:31:03	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:30:50	2018-09-17 16:02:01	13.1328829526901
853	2018-09-17 19:31:03	com.apple.carplay.oem	3	3	8.0	NULL	1.0	2018-09-17 19:30:50	2018-09-17 16:02:01	13.1328829526901
854	2018-09-17 19:31:14	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:31:01	2018-09-17 16:02:01	13.1328829526901
855	2018-09-17 19:31:32	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:31:19	2018-09-17 16:02:01	13.1328829526901
856	2018-09-17 19:31:32	com.apple.carplay.oem	3	3	8.0	NULL	1.0	2018-09-17 19:31:19	2018-09-17 16:02:01	13.1328829526901
857	2018-09-17 19:31:33	com.apple.Music	1	3	1.0	NULL	0.89	2018-09-17 19:31:20	2018-09-17 16:02:01	13.1328829526901
858	2018-09-17 19:34:53	com.apple.springboard.home-screen	1	3	0.0	NULL	1.0	2018-09-17 19:34:40	2018-09-17 16:02:01	13.1328829526901
859	2018-09-17 19:34:56	com.audible.iphone	1	3	1.0	NULL	1.0	2018-09-17 19:34:43	2018-09-17 16:02:01	13.1328829526901
860	2018-09-17 19:48:58	com.audible.iphone	1	3	1.0	NULL	1.0	2018-09-17 19:48:45	2018-09-17 16:02:01	13.1328829526901
861	2018-09-17 19:48:58	com.apple.Siri	4	3	3.0	NULL	1.0	2018-09-17 19:48:45	2018-09-17 16:02:01	13.1328829526901

macOS: `/private/var/db/powerlog/Library/BatteryLife/` (and `/Archives` directory)
- However, it appears macOS does not record this app usage data.

iOS: `/private/var/containers/Shared/SystemGroup/<GUID>/Library/BatteryLife/` (and `/Archives` directory)

This shows nearly the same data as `knowledgeC.db`, however it may show additional features such as CarPlay, home screen, and lock screen data.

The major caveat with the Powerlog is that some tables (not all!) will have a timestamp offset – testing is critical in this case. Never take a timestamp as an absolute. This particular query uses another table in the database to calculate the correct timestamp offsets.

References:
`https://www.mac4n6.com/blog/2018/12/16/on-the-third-day-of-apollo-my-true-love-gave-to-me-application-usage-to-determine-who-has-been-naughty-or-nice`
`https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series`
`https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie`

Health – Heart Rate – healthdb_secure.sqlite (iOS) [forever]

```
1   select datetime(samples.start_date+978307200,'unixepoch') as "Start Date",
2   datetime(samples.end_date+978307200,'unixepoch') as "End Date", samples.
3   data_type as "Data Type",
4   quantity,
5   original_quantity,
6   unit_strings.unit_string,
7   metadata_keys.key,
8   samples.data_id as "Samples Table ID"
9   from samples
10  left outer join quantity_samples on samples.data_id = quantity_samples.data_id
11  left outer join unit_strings on quantity_samples.original_unit = unit_strings.RowID
12  left outer join correlations on samples.data_id = correlations.object
13  left outer join metadata_values on metadata_values.object_id = samples.data_id
14  left outer join metadata_keys on metadata_keys.ROWID = metadata_values.key_id
15  where samples.data_type = 5
```

	Start Date	End Date	Data Type	quantity	original_quantity	unit_string	key	Samples Table ID
479576	2018-09-16 16:24:46	2018-09-16 16:24:46	5	3.01666666666667	181.0	count/min	_HKPrivateHeartRateContext	3637289
479577	2018-09-16 16:24:50	2018-09-16 16:24:50	5	3.03333333333333	182.0	count/min	HKMetadataKeyHeartRateMotionContext	3637294
479578	2018-09-16 16:24:50	2018-09-16 16:24:50	5	3.03333333333333	182.0	count/min	_HKPrivateHeartRateContext	3637294
479579	2018-09-16 16:24:53	2018-09-16 16:24:53	5	3.06666666666667	184.0	count/min	HKMetadataKeyHeartRateMotionContext	3637301
479580	2018-09-16 16:24:53	2018-09-16 16:24:53	5	3.06666666666667	184.0	count/min	_HKPrivateHeartRateContext	3637301
479581	2018-09-16 16:25:03	2018-09-16 16:25:03	5	3.11666666666667	187.0	count/min	HKMetadataKeyHeartRateMotionContext	3637310
479582	2018-09-16 16:25:03	2018-09-16 16:25:03	5	3.11666666666667	187.0	count/min	_HKPrivateHeartRateContext	3637310
479583	2018-09-16 16:25:07	2018-09-16 16:25:07	5	3.1	186.0	count/min	HKMetadataKeyHeartRateMotionContext	3637318
479584	2018-09-16 16:25:07	2018-09-16 16:25:07	5	3.1	186.0	count/min	_HKPrivateHeartRateContext	3637318
479585	2018-09-16 16:25:09	2018-09-16 16:25:09	5	3.08333333333333	185.0	count/min	HKMetadataKeyHeartRateMotionContext	3637335
479586	2018-09-16 16:25:09	2018-09-16 16:25:09	5	3.08333333333333	185.0	count/min	_HKPrivateHeartRateContext	3637335

SANS | DFIR FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 9

iOS: /private/var/mobile/Library/Health/

The health database is stored in an encrypted backup or available in a physical dump.

This query extracts the heart rate of the user, which is generated by an Apple Watch. This particular screenshot shows the author at the gym running (and regretting it!)

References:
https://www.mac4n6.com/blog/2018/12/15/on-the-second-day-of-apollo-my-true-love-gave-to-me-holiday-treats-and-a-trip-to-the-gym-a-look-at-ios-health-data
https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series
https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie

Health – Steps and Distance – healthdb_secure.sqlite (iOS) [forever]

```
1  select datetime(samples.start_date+978307200,'unixepoch','utc') as "Start Date",
2  datetime(samples.end_date+978307200,'unixepoch','utc') as "End Date",
3  samples.
4  data_type as "Data Type",
5  quantity as "Steps",
6  samples.data_id as "Samples Table ID"
7  from samples
8  left outer join quantity_samples on samples.data_id = quantity_samples.data_id
9  left outer join unit_strings on quantity_samples.original_unit = unit_strings.RowID
10 left outer join correlations on samples.data_id = correlations.object
11 left outer join metadata_values on metadata_values.object_id = samples.data_id
12 left outer join metadata_keys on metadata_keys.ROWID = metadata_values.key_id
13 where samples.data_type = 7 and key is null
14 AND "START DATE" LIKE '%2018-09-16%'
15 ORDER BY "Start Date"
```

	Start Date	End Date	Data Type	Steps	Samples Table ID
168	2018-09-16 20:18:01	2018-09-16 20:19:02	7	133.0	3637721
169	2018-09-16 20:19:02	2018-09-16 20:20:04	7	169.0	3637722
170	2018-09-16 20:20:04	2018-09-16 20:21:05	7	138.0	3637723
171	2018-09-16 20:21:05	2018-09-16 20:22:06	7	111.0	3637724
172	2018-09-16 20:22:06	2018-09-16 20:23:08	7	168.0	3637725
173	2018-09-16 20:23:08	2018-09-16 20:24:09	7	114.0	3637726
174	2018-09-16 20:24:09	2018-09-16 20:25:48	7	131.0	3637727
175	2018-09-16 20:24:54	2018-09-16 20:34:52	7	442.0	3637711
176	2018-09-16 20:25:48	2018-09-16 20:28:38	7	14.0	3637728
177	2018-09-16 20:29:47	2018-09-16 20:30:48	7	33.0	3637729
178	2018-09-16 20:30:48	2018-09-16 20:40:11	7	1018.0	3637785
179	2018-09-16 20:34:52	2018-09-16 20:43:01	7	699.0	3637712

```
1  select datetime(samples.start_date+978307200,'unixepoch','utc') as "Start Date",
2  datetime(samples.end_date+978307200,'unixepoch','utc') as "End Date",
3  samples.data_type as "Data Type",
4  quantity as "Distance in Meters",
5  samples.data_id as "Samples Table ID"
6  from samples
7  left outer join quantity_samples on samples.data_id = quantity_samples.data_id
8  left outer join unit_strings on quantity_samples.original_unit = unit_strings.RowID
9  left outer join correlations on samples.data_id = correlations.object
10 left outer join metadata_values on metadata_values.object_id = samples.data_id
11 left outer join metadata_keys on metadata_keys.ROWID = metadata_values.key_id
12 where samples.data_type = 8 and key is null
13 AND "START DATE" LIKE '%2018-09-16%'
14 ORDER BY "Start Date"
```

	Start Date	End Date	Data Type	Distance in Meters	Samples Table ID
150	2018-09-16 19:25:30	2018-09-16 19:35:23	8	78.6573735140264	3634387
151	2018-09-16 19:35:23	2018-09-16 19:45:20	8	416.445916654542	3634938
152	2018-09-16 19:45:20	2018-09-16 19:55:18	8	238.186128458008	3635697
153	2018-09-16 19:55:18	2018-09-16 20:05:16	8	200.604227676522	3636274
154	2018-09-16 20:05:16	2018-09-16 20:15:15	8	700.755024724873	3636847
155	2018-09-16 20:15:15	2018-09-16 20:25:07	8	1113.33662894019	3637479
156	2018-09-16 20:24:54	2018-09-16 20:34:52	8	299.481329584727	3637996
157	2018-09-16 20:25:07	2018-09-16 20:26:08	8	6.43414319492877	3637716
158	2018-09-16 20:29:47	2018-09-16 20:30:48	8	21.5141206183471	3637717
159	2018-09-16 20:30:48	2018-09-16 20:40:11	8	658.256004666211	3637784
160	2018-09-16 20:34:52	2018-09-16 20:43:01	8	502.089796816697	3637997
161	2018-09-16 20:40:11	2018-09-16 20:43:09	8	38.4893803959712	3637786
162	2018-09-16 21:09:58	2018-09-16 21:11:00	8	14.7879310355056	3637795

iOS: /private/var/mobile/Library/Health/

The health database is stored in an encrypted backup or available in a physical dump.

These two queries are extracting the steps and distance from the health database. These can be useful in determining how fast a user was going or if they were moving at all.

References:
https://www.mac4n6.com/blog/2018/12/15/on-the-second-day-of-apollo-my-true-love-gave-to-me-holiday-treats-and-a-trip-to-the-gym-a-look-at-ios-health-data
https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series

Device Status – knowledgeC.db – Now Playing

	START	END	USAGE IN SECONDS	BUNDLE ID	NOW PLAYING ALBUM	NOW PLAYING ARTIST	NOW PLAYING GENRE	NOW PLAYING TITLE	NOW PLAYING DURATION	STREAM NAME
238	2018-08-28 09:12:07	2018-08-28 09:12:48	41	com.apple.Music	Nervous System	Julia Michaels	Pop	Issues	176.378775510204	/media/nowPlaying
239	2018-08-28 18:52:49	2018-08-28 19:32:09	2360	org.npr.nprnews	WHRV • Live	NULL	NULL	WHRV-FM NPR for Eastern Virginia	0.0	/media/nowPlaying
240	2018-08-28 19:32:12	2018-08-28 19:32:12	0	org.npr.nprnews	WHRV • Live	NULL	NULL	WHRV-FM NPR for Eastern Virginia	0.0	/media/nowPlaying
241	2018-08-28 19:35:51	2018-08-28 19:35:52	1	org.npr.nprnews	WHRV • Live	NULL	NULL	WHRV-FM NPR for Eastern Virginia	0.0	/media/nowPlaying
242	2018-08-28 19:35:52	2018-08-28 19:39:58	246	com.apple.Music	Love Stuff	Elle King	Alternative	America's Sweetheart	245.542	/media/nowPlaying
243	2018-08-28 19:39:58	2018-08-28 19:40:03	5	com.apple.Music	Evacuate the Dancefloor	Cascada	Dance	Evacuate the Dancefloor	207.865034013605	/media/nowPlaying
244	2018-08-28 19:40:03	2018-08-28 19:40:05	2	com.apple.Music	With Teeth	Nine Inch Nails	Rock	The Hand That Feeds	211.789206349206	/media/nowPlaying
245	2018-08-28 19:40:05	2018-08-28 19:40:06	1	com.apple.Music	7/27 (Deluxe)	Fifth Harmony	Pop	Work from Home (feat. Ty Dolla $i...	214.529160997732	/media/nowPlaying
246	2018-08-28 19:40:06	2018-08-28 19:40:07	1	com.apple.Music	Nice to Meet You - EP	Seeb & Dagny	Pop	Drink About	182.323083900227	/media/nowPlaying
247	2018-08-28 19:40:07	2018-08-28 19:41:39	92	com.apple.Music	Blackout (feat. Steph Jones) ...	Tritonal	Dance	Blackout (feat. Steph Jones)	211.742766439909	/media/nowPlaying
248	2018-08-28 19:41:39	2018-08-29 04:15:40	30841	com.apple.Music	Blackout (feat. Steph Jones) ...	Tritonal	Dance	Blackout (feat. Steph Jones)	211.742766439909	/media/nowPlaying
249	2018-08-29 04:15:40	2018-08-29 04:15:43	3	com.apple.Music	Blackout (feat. Steph Jones) ...	Tritonal	Dance	Blackout (feat. Steph Jones)	211.69	/media/nowPlaying
250	2018-08-29 04:15:44	2018-08-29 04:19:16	212	com.apple.Music	Blackout (feat. Steph Jones) ...	Tritonal	Dance	Blackout (feat. Steph Jones)	211.69	/media/nowPlaying
251	2018-08-29 04:26:42	2018-08-29 04:26:42	0	com.apple.Music	Alex Rider: Operation Stormb...	Curve	Soundtrack	Chinese Burn (Lunatic Calm Mix)	445.637369614512	/media/nowPlaying
252	2018-08-29 04:26:42	2018-08-29 04:30:24	222	com.apple.Music	Lest We Forget: The Best of ...	Marilyn Manson	Rock	The Beautiful People	222.841904761905	/media/nowPlaying
253	2018-08-29 04:30:25	2018-08-29 04:34:02	217	com.apple.Music	Version 2.0 (20th Anniversar...	Garbage	Rock	I Think I'm Paranoid	218.1979138322	/media/nowPlaying
254	2018-08-29 04:37:54	2018-08-29 04:37:54	0	com.apple.Music	Funk Wav Bounces Vol. 1	Calvin Harris	Dance	Slide (feat. Frank Ocean & Migos)	230.876009070295	/media/nowPlaying

macOS: ~/Library/Application Support/Knowledge/knowledgeC.db
iOS: /private/var/mobile/Library/CoreDuet/Knowledge/knowledgeC.db

The knowledgeC.db keeps track of what is playing and how it is playing.

References:
https://www.mac4n6.com/blog/2018/8/5/knowledge-is-power-using-the-knowledgecdb-database-on-macos-and-ios-to-determine-precise-user-and-application-usage
https://www.mac4n6.com/blog/2018/9/12/knowledge-is-power-ii-a-day-in-the-life-of-my-iphone-using-knowledgecdb
https://www.mac4n6.com/blog/2018/12/17/on-the-fourth-day-of-apollo-my-true-love-gave-to-me-media-analysis-to-prove-you-listened-to-all-i-want-for-christmas-is-you-over-and-over-since-before-thanksgiving
https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series
https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie

Device Status – knowledgeC.db – Locked, Plugged In?

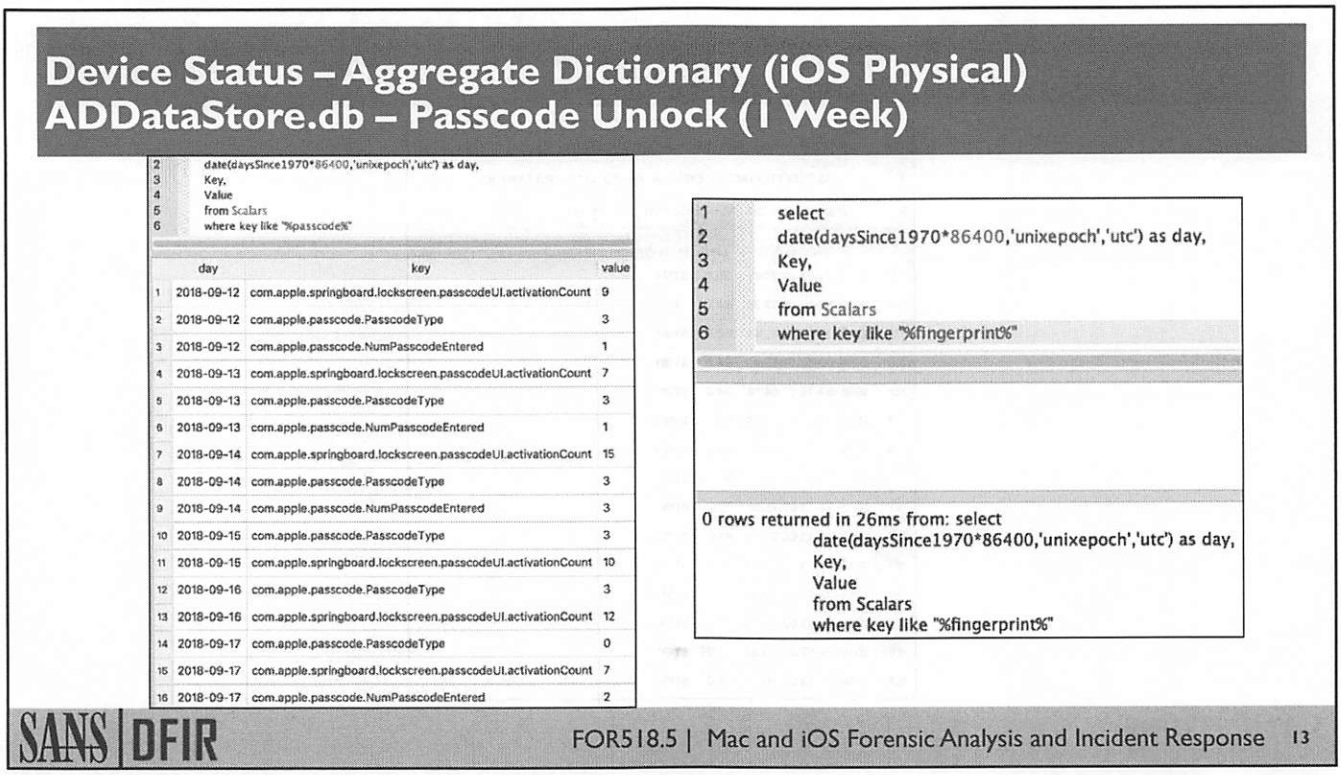
	ENTRY CREATION	DAY OF WEEK	START	END	USAGE IN SECONDS	IS LOCKED	STREAM NAME	ZOBJECT TABLE ID
57	2018-08-22 04:41:45	Tuesday	2018-08-22 04:39:48	2018-08-22 04:41:44	116	UNLOCKED	/device/isLocked	296308
58	2018-08-22 04:48:53	Tuesday	2018-08-22 04:41:44	2018-08-22 04:48:52	428	LOCKED	/device/isLocked	296314
59	2018-08-22 04:49:22	Tuesday	2018-08-22 04:48:52	2018-08-22 04:49:20	28	UNLOCKED	/device/isLocked	296317
60	2018-08-22 04:50:09	Tuesday	2018-08-22 04:49:20	2018-08-22 04:50:08	48	LOCKED	/device/isLocked	296319
61	2018-08-22 04:52:02	Tuesday	2018-08-22 04:50:08	2018-08-22 04:52:00	112	UNLOCKED	/device/isLocked	296322

	ENTRY CREATION	DAY OF WEEK	START	END	USAGE IN SECONDS	IS PLUGGED IN	STREAM NAME	ZOBJECT TABLE ID
67	2018-09-01 05:29:23	Friday	2018-09-01 05:26:44	2018-09-01 05:29:20	156	PLUGGED IN	/device/isPluggedIn	307290
68	2018-09-01 10:08:25	Friday	2018-09-01 05:29:20	2018-09-01 10:08:24	16744	UNPLUGGED	/device/isPluggedIn	307699
69	2018-09-01 15:52:47	Friday	2018-09-01 10:08:24	2018-09-01 15:52:44	20660	PLUGGED IN	/device/isPluggedIn	307792
70	2018-09-01 16:25:30	Saturday	2018-09-01 15:52:44	2018-09-01 16:25:28	1964	UNPLUGGED	/device/isPluggedIn	307816
71	2018-09-01 21:09:29	Saturday	2018-09-01 16:25:28	2018-09-01 21:09:28	17040	PLUGGED IN	/device/isPluggedIn	307876
72	2018-09-02 11:10:56	Saturday	2018-09-01 21:09:28	2018-09-02 11:10:56	50488	UNPLUGGED	/device/isPluggedIn	308178

macOS: ~/Library/Application Support/Knowledge/knowledgeC.db
iOS: /private/var/mobile/Library/CoreDuet/Knowledge/knowledgeC.db

The knowledgeC.db keeps track when a device is locked and when it is plugged in.

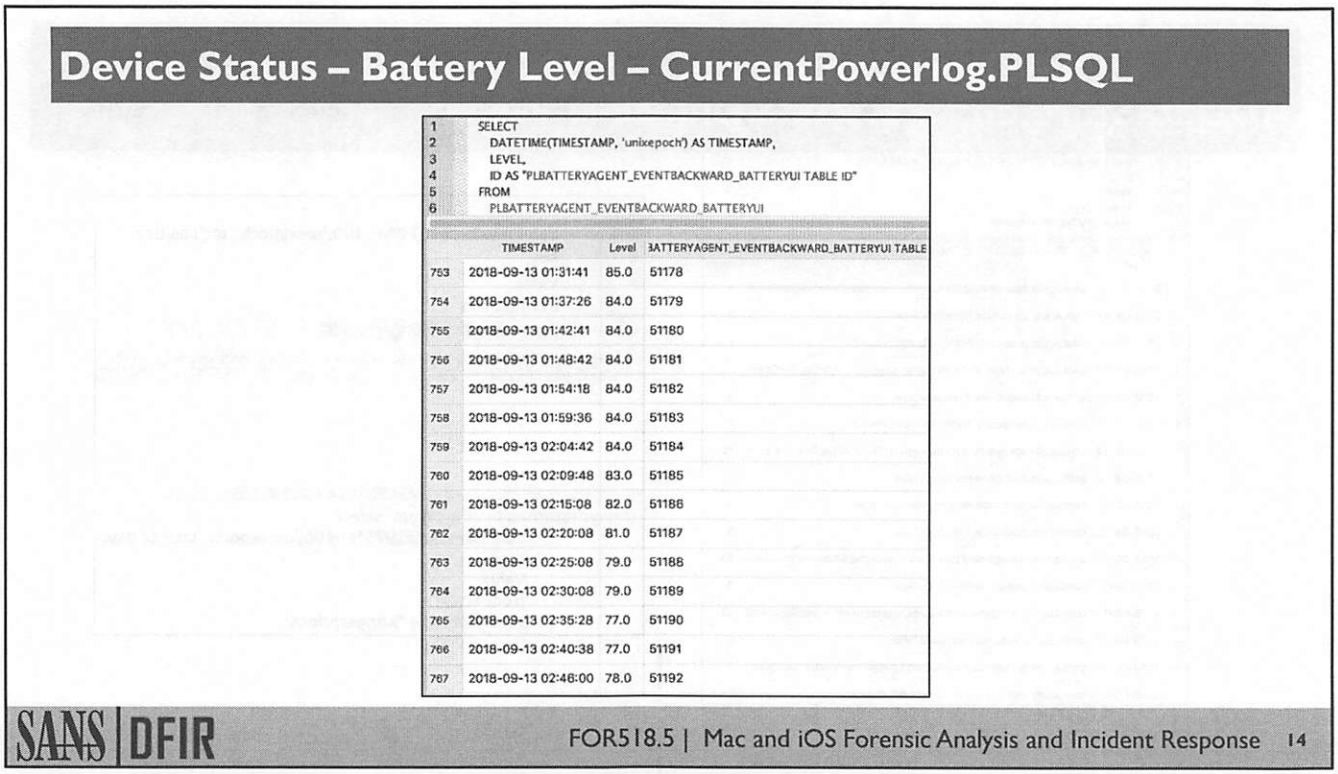
References:
https://www.mac4n6.com/blog/2018/8/5/knowledge-is-power-using-the-knowledgecdb-database-on-macos-and-ios-to-determine-precise-user-and-application-usage
https://www.mac4n6.com/blog/2018/9/12/knowledge-is-power-ii-a-day-in-the-life-of-my-iphone-using-knowledgecdb
https://www.mac4n6.com/blog/2018/12/21/on-the-eighth-day-of-apollo-my-true-love-gave-to-me-a-glorious-lightshow-analysis-of-device-connections
https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series
https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie



iOS: /private/var/mobile/Library/AggregateDictionary/ADDataStore.sqlitedb

The Aggregate Dictionary stores many different items, one of which is the methods used to access the iOS device. The left screenshot shows the passcode types and how many times it was used. On the left is an example of how hardware can show (or not show) different pieces of data. This data came from an iPhone X with no Touch ID – if the device had Touch ID it would show a few keys about how that feature was used.

References:
`https://www.mac4n6.com/blog/2017/3/12/introduction-to-the-aggregate-dictionary-database-addatastoresqlite`
`https://www.mac4n6.com/blog/2018/12/18/on-the-fifth-day-of-apollo-my-true-love-gave-to-me-a-stocking-full-of-random-junk-some-of-which-might-be-useful`
`https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series`
`https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie`



macOS: /private/var/db/powerlog/Library/BatteryLife/ (and /Archives directory
iOS: /private/var/containers/Shared/SystemGroup/<GUID>/Library/BatteryLife/ (and /Archives directory)

The powerlog keeps track of the battery status for Mac and iOS devices.

References:
`https://www.mac4n6.com/blog/2018/12/19/on-the-sixth-day-of-apollo-my-true-love-gave-to-me-blinky-things-with-buttons-device-status-analysis`
`https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series`
`https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie`

APOLLO: Apple Pattern of Life Lazy Output'er

Low Bar to Entry – Easy contribution for busy forensic investigators

- Almost anyone can develop a SQL query

Quick Correlation – Not perfect, but works!

Easy to update and configure SQL queries

- Across devices and across OS versions

SQL Script Sharing

Dumb Script – Nothin' fancy going on here, the real work is done by the modules

- Pro Tip: It can be used with any SQL database and query… Android, Windows… even Blackberry!?

Modules Directory – SQL query configuration files

References:

https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series

https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie

APOLLO Output Example – Distracted Driving?

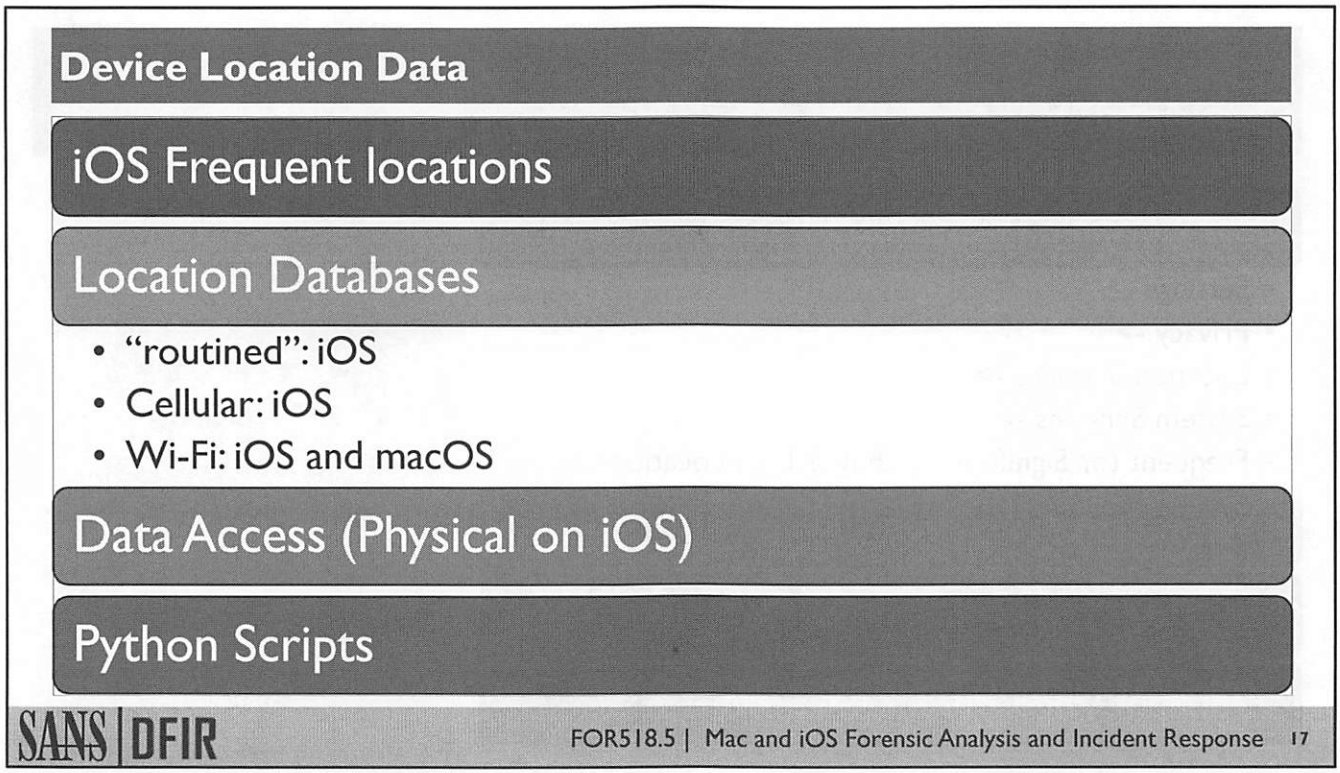
	Activity	Output
316	Location	[TIMESTAMP: 2018-09-16 04:03:48] [COORDINATES: 39.1153389157401, -76.6329456498556] [ALTITUDE: 30.4] [COURSE: 151.890029907] [SPEED: 15.8448888889] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
317	Location	[TIMESTAMP: 2018-09-16 04:03:49] [COORDINATES: 39.1152105275425, -76.632857956339] [ALTITUDE: 30.5] [COURSE: 152.069168091] [SPEED: 15.9477777778] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURA...
318	Location	[TIMESTAMP: 2018-09-16 04:03:50] [COORDINATES: 39.1150827819936, -76.6327722606686] [ALTITUDE: 30.7] [COURSE: 152.069168091] [SPEED: 15.9477777778] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURA...
319	App Usage	[ZINTERACTIONS CREATION DATE: 2018-09-16 04:03:52] [ZBUNDLEID: com.apple.assistant_service] [ZDISPLAYNAME: None] [ZIDENTIFIER: None] [ZPERSONID: None] [ZDIRECTION: 3] [ZISRESPONSE: 0] [ZMECHANISM:...
320	App Usage	[ZINTERACTIONS CREATION DATE: 2018-09-16 04:03:52] [ZBUNDLEID: com.apple.MobileSMS] [ZDISPLAYNAME: None] [ZIDENTIFIER: None] [ZPERSONID: None] [ZDIRECTION: 1] [ZISRESPONSE: 0] [ZMECHANISM: 4] [Z...
321	Location	[TIMESTAMP: 2018-09-16 04:03:51] [COORDINATES: 39.1149556073617, -76.6326878872086] [ALTITUDE: 30.8] [COURSE: 152.069244385] [SPEED: 16.0506666667] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
322	Location	[TIMESTAMP: 2018-09-16 04:03:52] [COORDINATES: 39.1148244918474, -76.6326033291797] [ALTITUDE: 30.8] [COURSE: 152.069168091] [SPEED: 15.9992222222] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
323	Application Activity	[ENTRY CREATION: 2018-09-16 04:03:52] [DAY OF WEEK: Sunday] [START: 2018-09-16 04:03:51] [END: 2018-09-16 04:03:51] [USAGE IN SECONDS: 0] [APP NAME: Messages] [INTENT CLASS: INSendMessageIntent] [IN...
324	Location	[TIMESTAMP: 2018-09-16 04:03:53] [COORDINATES: 39.1147407013582, -76.632554927344] [ALTITUDE: 30.8] [COURSE: 152.069168091] [SPEED: 15.6391111111] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURACY...
325	Application Usage	[ADJUSTED_TIMESTAMP: 2018-09-16 04:03:53] [BUNDLE_ID: com.apple.MobileSMS] [APPROLE: 1] [DISPLAY: 3] [LEVEL: 1.0] [ORIENTATION: None] [SCREENWEIGHT: 0.89] [ORIGINAL_SCREEN_STATE_TIMESTAMP: 2018...
326	Location	[TIMESTAMP: 2018-09-16 04:03:54] [COORDINATES: 39.1145646633985, -76.6324465294773] [ALTITUDE: 30.7] [COURSE: 156.159225464] [SPEED: 15.6391111111] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURA...
327	Application Activity	[ENTRY CREATION: 2018-09-16 04:03:54] [DAY OF WEEK: Sunday] [START: 2018-09-16 04:03:15] [END: 2018-09-16 04:03:54] [USAGE IN SECONDS: 39] [BUNDLE ID: com.apple.Music] [NOW PLAYING ALBUM: Sugar] [...
328	Device Status	[ENTRY CREATION: 2018-09-16 04:03:54] [DAY OF WEEK: Sunday] [START: 2018-09-16 04:03:12] [END: 2018-09-16 04:03:52] [USAGE IN SECONDS: 40] [AUDIO IDENTIFIER: 74:6F:F7:20:6D:77-Audio-AudioMain-43483...
329	Location	[TIMESTAMP: 2018-09-16 04:03:55] [COORDINATES: 39.114432869852, -76.6323719029567] [ALTITUDE: 30.4] [COURSE: 156.159301758] [SPEED: 16.1021111111] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURACY...
330	App Usage	[TIMESTAMP: 2018-09-16 04:03:55] [TIMESTAMP LOGGED: 2018-09-16 04:03:55] [APPLICATION NAME / BUNDLE ID: assistantd] [ASERTION ID: 38188] [ASSERTION NAME: com.apple.audio.sid:0x379e0cb, assistantd(14...
331	App Usage	[TIMESTAMP: 2018-09-16 04:03:55] [TIMESTAMP LOGGED: 2018-09-16 04:03:55] [APPLICATION NAME / BUNDLE ID: com.apple.Music] [ASERTION ID: 38320] [ASSERTION NAME: com.apple.audio.sid:0x379e4db, Music(...
332	Location	[TIMESTAMP: 2018-09-16 04:03:56] [COORDINATES: 39.1142981795731, -76.6322986162874] [ALTITUDE: 30.2] [COURSE: 157.228683472] [SPEED: 16.5136666667] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCURA...

References:

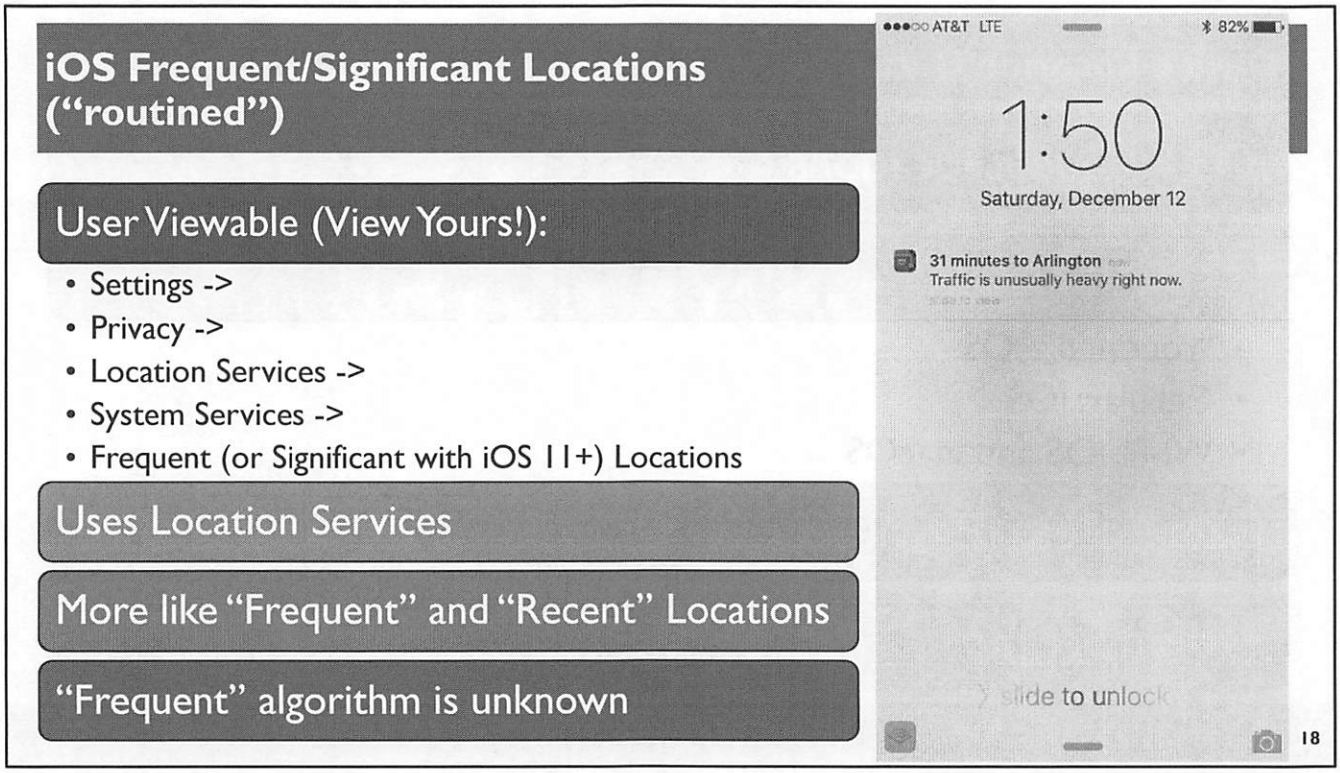
https://www.mac4n6.com/blog/2018/12/14/on-the-first-day-of-apollo-my-true-love-gave-to-me-a-python-script-an-introduction-to-the-apple-pattern-of-life-lazy-outputer-apollo-blog-series

https://github.com/mac4n6/Presentations/tree/master/From%20Apple%20Seeds%20to%20Apple%20Pie

https://www.mac4n6.com/blog/2018/12/24/on-the-eleventh-day-of-apollo-my-true-love-gave-to-me-an-intriguing-story-putting-it-all-together-a-day-in-the-life-of-my-iphone-using-apollo



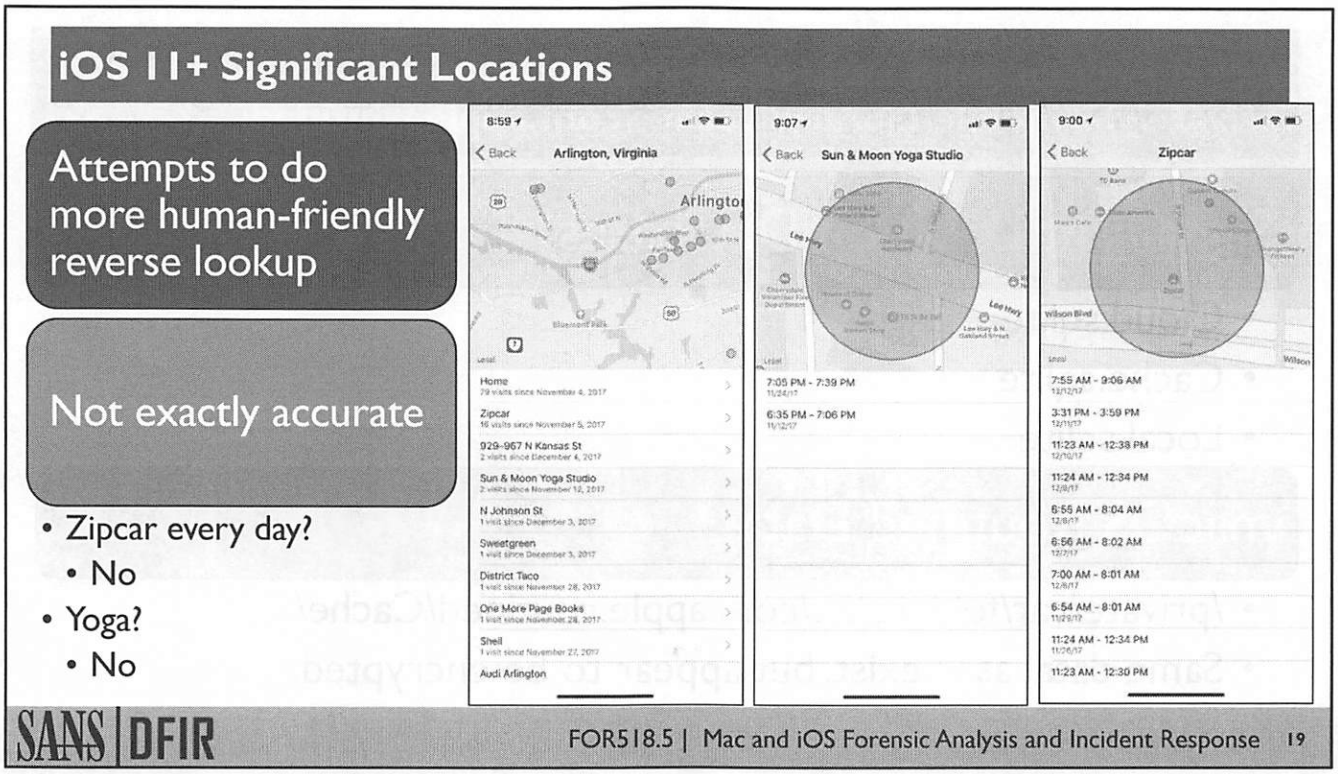
iOS location data can be found in a variety of databases and files on the physical file system. The files discussed in this next section will only be available on physical acquisitions.



The "routined" process on iOS devices keeps track of the user's routine. As shown in the screenshot above, my iPhone was able to determine when I was headed back home to Arlington, and popped up an alert to tell me that traffic was unusually heavy at that moment.

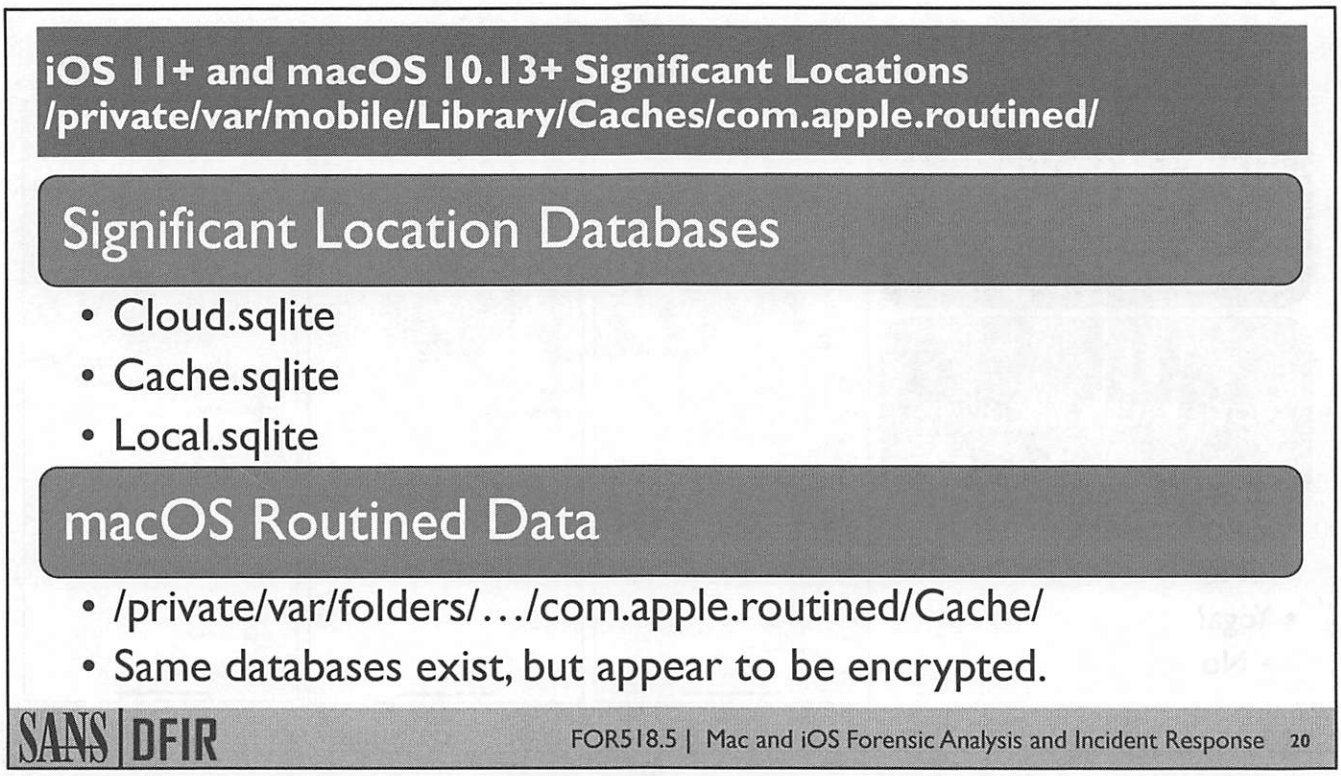
This process keeps track of a device's location and finds routines in their pattern. This location data is stored in the Frequent Locations area of Location Services Settings:
`Settings | Privacy | Location Services | System Services | Frequent (or Significant) Locations`

If enabled, this area will show a user's frequent and recent locations. While all locations are tracked, only those that get bubbled at the top of "frequent" or "recent" are shown in this list. This is not an inclusive list of everywhere the user has been.



iOS 11 started attempting a more "human-friendly" reverse location lookup; however, you should be careful how you use this data. It can be close but quite wrong. In the example above (found in my own data), two entries jumped out at me: Zipcar and a yoga studio.

I definitely do not visit the Zipcar every day (granted some folks might)—I'm at the Orangetheory gym nearby! As far as the yoga goes, I'm much more comfortable in the ramen restaurant across the street!



A completely new format for the routine data has been introduced in iOS 11.

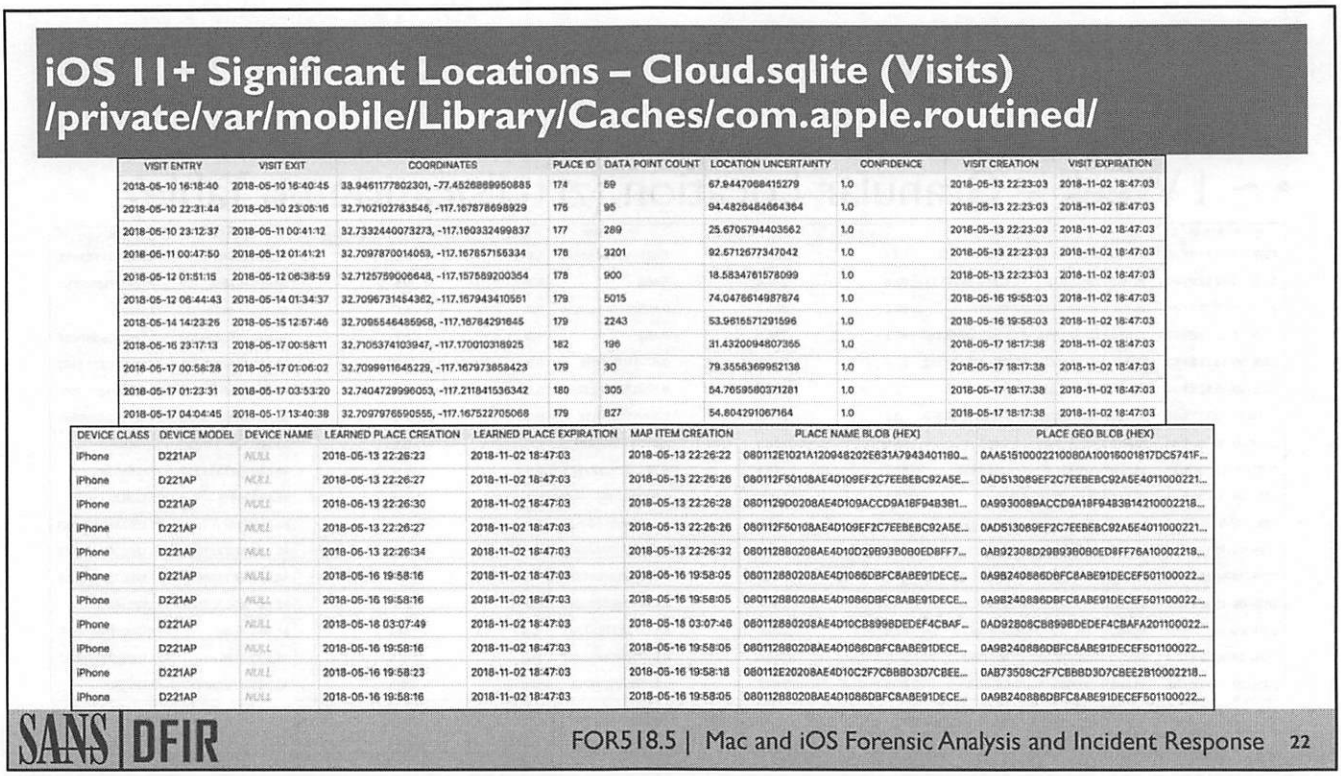
This data may also be found on macOS 10.13+; however, it appears encrypted.

iOS 11+ Significant Locations – Cache.sqlite /private/var/mobile/Library/Caches/com.apple.routined/

- ~ 1 Week – Granular Location (zrtcllocationmo Table)

TIMESTAMP	COORDINATES	ALTITUDE	COURSE	SPEED (M/S)	HORIZONTAL ACCURACY	VERTICAL ACCURACY	LATITUDE	LONGITUDE
2018-09-10 12:51:39	38.8805329833333, -77.1250671833333	82.3	238.6	7.92244444444445	5.0	9.5	38.8805329833333	-77.1250671833333
2018-09-10 12:61:40	38.8804976333333, -77.1251412833333	82.3	239.0	7.408	5.0	9.5	38.8804976333333	-77.1251412833333
2018-09-10 12:51:41	38.8804646833333, -77.125211	82.2	239.3	6.89355555555556	5.0	9.5	38.8804646833333	-77.125211
2018-09-10 12:51:42	38.8804345833333, -77.1252758666667	82.2	239.8	6.482	5.0	9.5	38.8804345833333	-77.1252758666667
2018-09-10 12:51:43	38.8804052166667, -77.1253403333333	82.2	240.2	6.37911111111111	5.0	9.5	38.8804052166667	-77.1253403333333
2018-09-10 12:51:44	38.8803767, -77.1254038666667	82.2	240.5	6.27622222222222	5.0	9.5	38.8803767	-77.1254038666667
2018-09-10 12:51:45	38.8803483833333, -77.1254684166667	82.2	241.3	6.37911111111111	5.0	9.5	38.8803483833333	-77.1254684166667
2018-09-10 12:51:46	38.8803202333333, -77.1255354	82.2	242.5	6.63633333333333	5.0	9.5	38.8803202333333	-77.1255354
2018-09-10 12:51:47	38.8802927333333, -77.12560385	82.2	243.4	6.73922222222222	5.0	9.5	38.8802927333333	-77.12560385
2018-09-10 12:51:48	38.88026615, -77.125672	82.1	244.1	6.53444444444444	5.0	9.5	38.88026615	-77.125672
2018-09-10 12:51:49	38.8802407, -77.1257390833333	82.1	244.4	6.43055555555556	5.0	9.5	38.8802407	-77.1257390833333
2018-09-10 12:51:50	38.8802163333333, -77.1258067333333	82.1	244.6	6.58488888888889	5.0	9.5	38.8802163333333	-77.1258067333333
2018-09-10 12:51:51	38.8801892666667, -77.1258767833333	82.0	244.8	6.79066666666667	5.0	9.5	38.8801892666667	-77.1258767833333
2018-09-10 12:51:52	38.8801638666667, -77.12594635	82.0	245.4	6.58488888888889	5.0	9.5	38.8801638666667	-77.12594635
2018-09-10 12:51:53	38.88014065, -77.1260116666667	82.0	246.0	6.17333333333333	10.0	19.0	38.88014065	-77.1260116666667
2018-09-10 12:51:54	38.88011795, -77.1260777666667	81.9	246.6	6.32766666666667	10.0	19.0	38.88011795	-77.1260777666667
2018-09-10 12:51:55	38.8800932666667, -77.1261517	81.9	247.2	7.15077777777778	10.0	19.0	38.8800932666667	-77.1261517

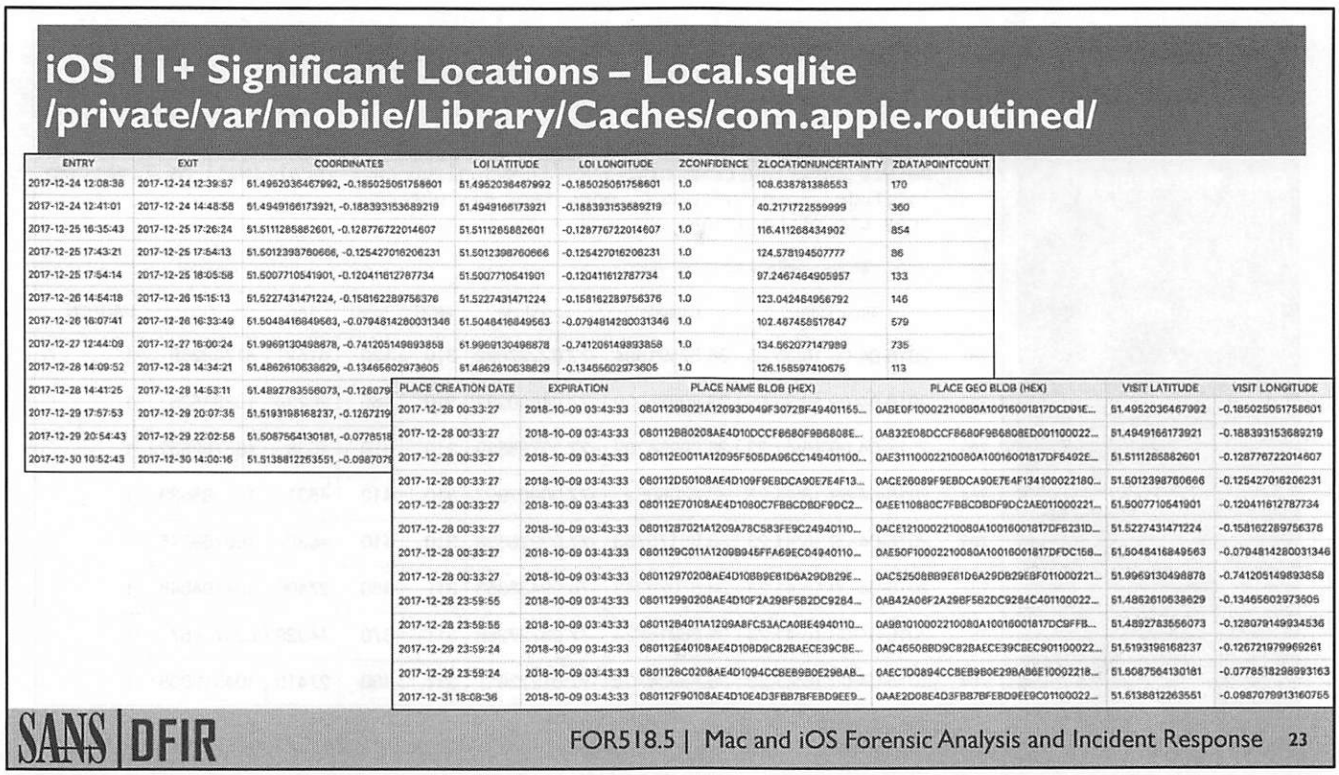
An example of the data found in the Cache.sqlite database is shown above. This may contain very granular location data for the device for about a week. This data was extracted using the routined_cache_zrtcllocationmo query from the APOLLO framework:
`https://github.com/mac4n6/APOLLO/blob/master/modules/routined_cache_zrtcllocationmo.txt`



This example from the Cloud.sqlite database shows visits to a certain location. This data was extracted using the routined_cloud_visit_entry module from the APOLLO framework. The extracted data is long enough that it needed two screenshots!

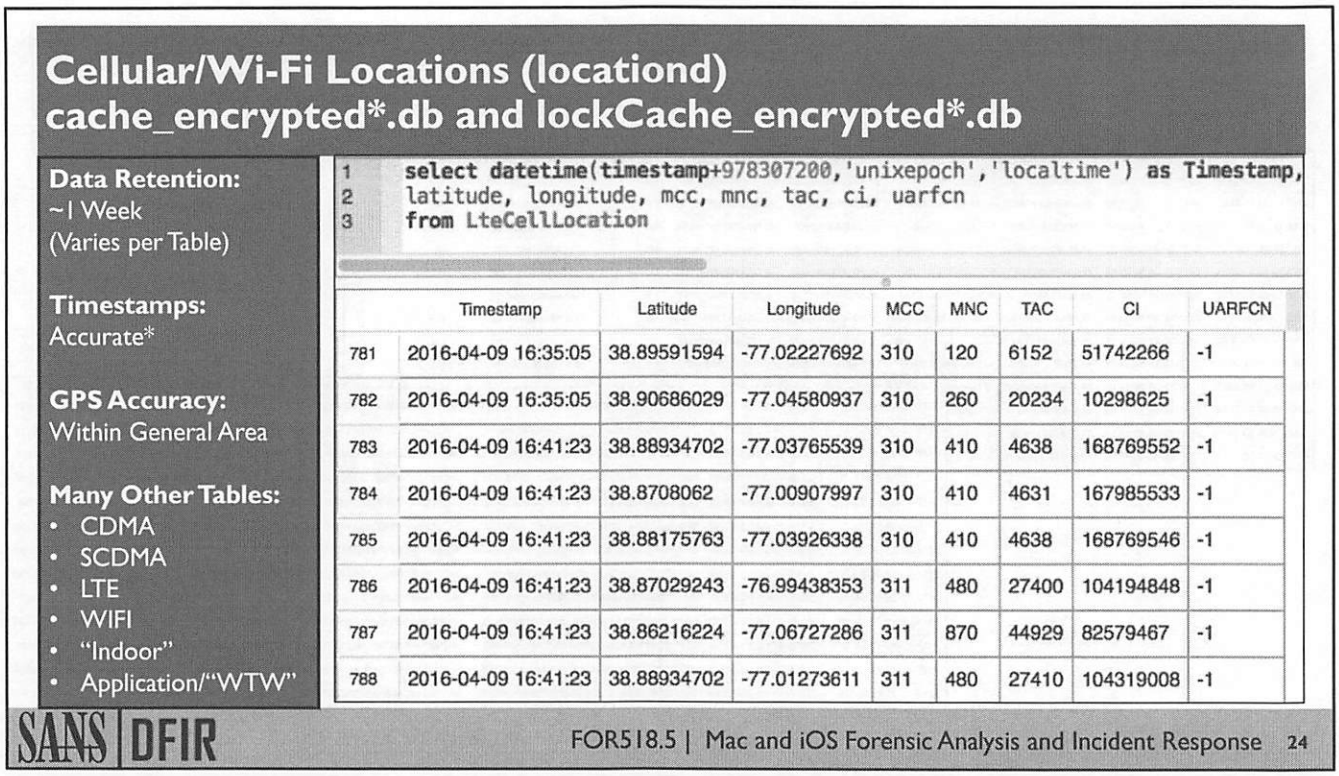
The data BLOBs in "Place Name" and "Place GEO BLOB" are location protobuf BLOBs.

Reference:
`https://github.com/mac4n6/APOLLO/blob/master/modules/routined_cloud_visit_entry.txt`



Similar to the previous examples, this data comes from the Local.sqlite database. Again, the data extracted takes up two screenshots.

This is Locations of Interest data (Significant Locations). This data was extracted using the routined_local_learned_location_of_interest_entry module located here in the APOLLO framework: `https://github.com/mac4n6/APOLLO/blob/master/modules/routined_local_learned_location_of_interest_entry.txt`



iOS Physical: `/private/var/root/library/caches/locationd/cache_encrypted*.db` and `lockCache_encrypted*.db`

Along with "`routined`" or Frequent Locations, the location of various cellular and Wi-Fi access is recorded in a few databases on iOS. These databases are also only available on Physical/Physical-Logical acquisitions.

Depending on the cellular service (LTE/CDMA/SCDMA), this data may be stored in a different table. LTE was shown for the test data provided (`LteCellLocation` table). This table contains a timestamp in Mac Epoch, location coordinates, and various cellular data (MCC,MNC,TAC,CI,UARFCN). The timestamps are accurate, and the data is kept for about a week.

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

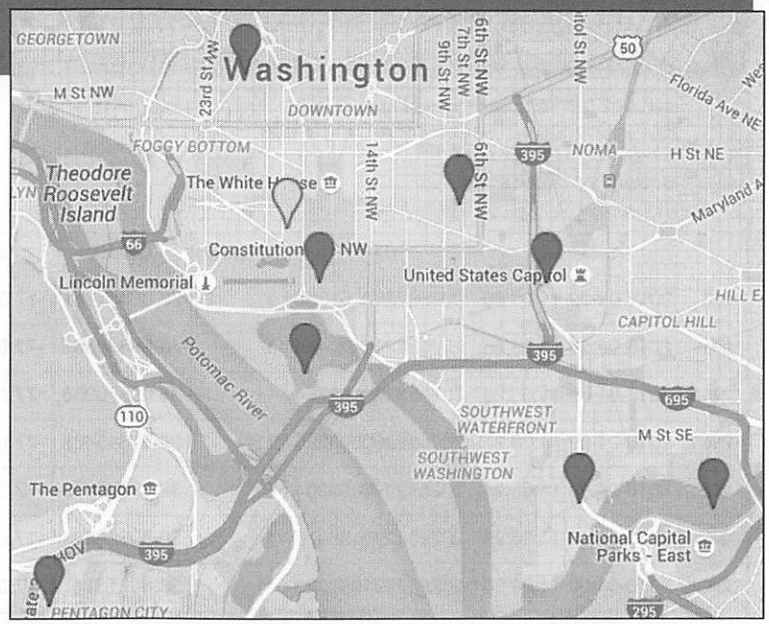
Reference:
`https://www.mac4n6.com/blog/2016/6/6/new-script-ios-locations-scraper`

Cellular Locations
cache_encryptedA.db

Yellow Point:
My Actual Location

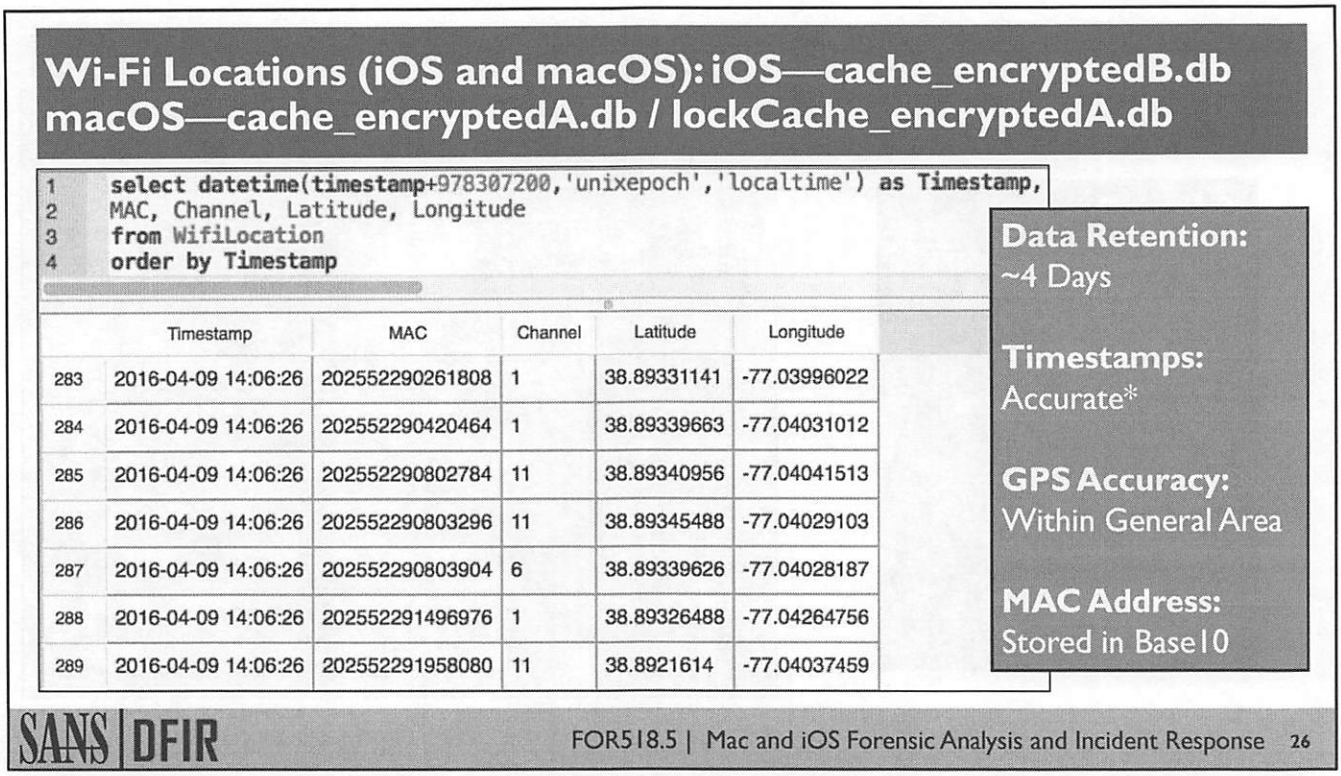
Red Points:
"LTE Cell Locations"

Warning:
Locations are in general area, NOT exact area



The cellular locations can also be mapped out; however, keep in mind that the locations are not of the device but of the cellular towers it may have been communicating with.

The yellow point right near the White House is where the device actually was at that instant, while the other locations are cellular towers. You can get a general idea of where the device was within the last week, but not an exact location.



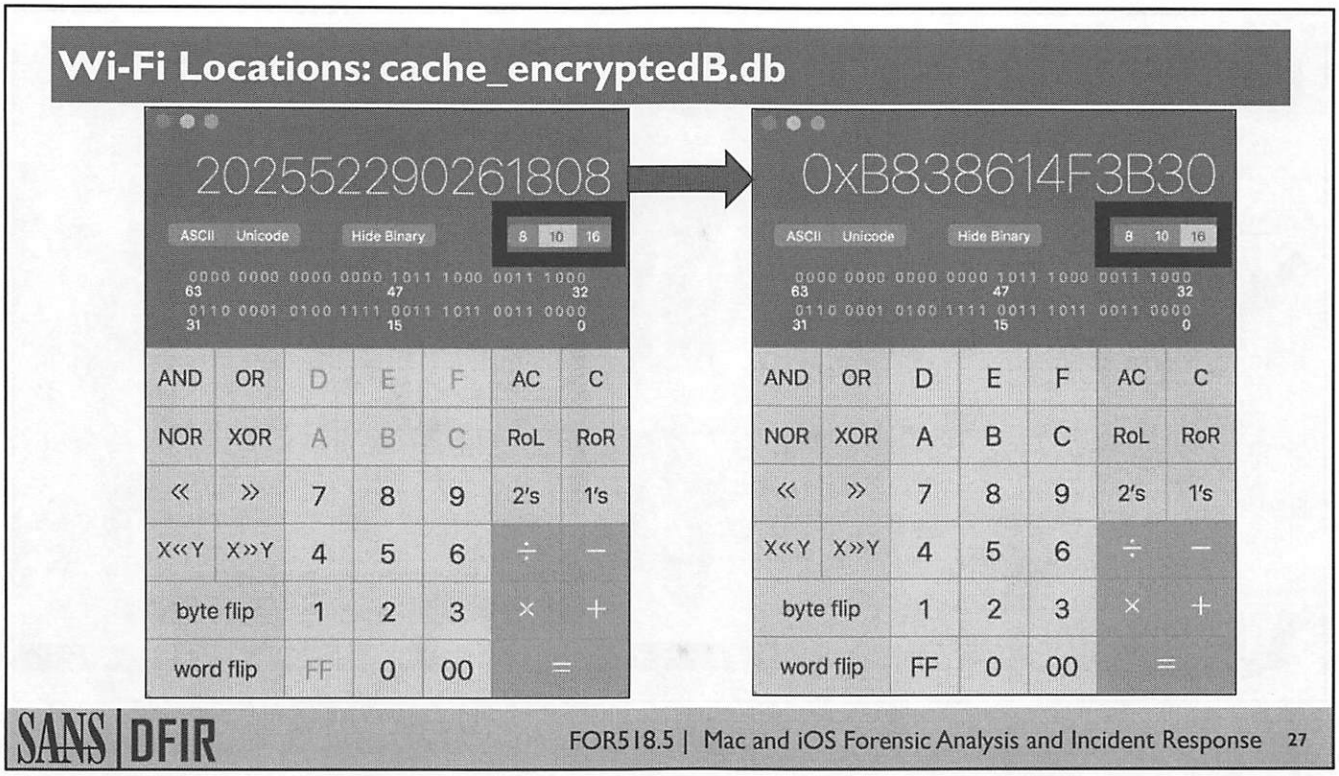
iOS Physical: `/private/var/root/library/caches/locationd/cache_encryptedB.db`
macOS:
`/private/var/folders/zz/zyxvpxvq6csfxvn_n00000sm00006d/cache_encryptedA.db` or `lockCache_encryptedA.db` (Directory names may differ.)

Along with "`routined`" locations, Wi-Fi locations are also pushed into this database in the `WifiLocation` table. The timestamp, MAC address, Wi-Fi Channel, and coordinates were extracted from this table. This data is kept for about four days and has accurate timestamps and location data. This data is most similar to that of wardriving data. Each iDevice is collecting this information to improve location accuracy for other devices. The user does not have to connect to any of the access points for this data to be collected—its collection is transparent to the user. It is worth noting that the access point's MAC address is stored in Base10 in this table.

The macOS version of the database appears to use the "normal" notation of hex for the MAC address. This Mac version also has a longer retention period.

The rows in this table have been ordered by the "`Timestamp`" column.

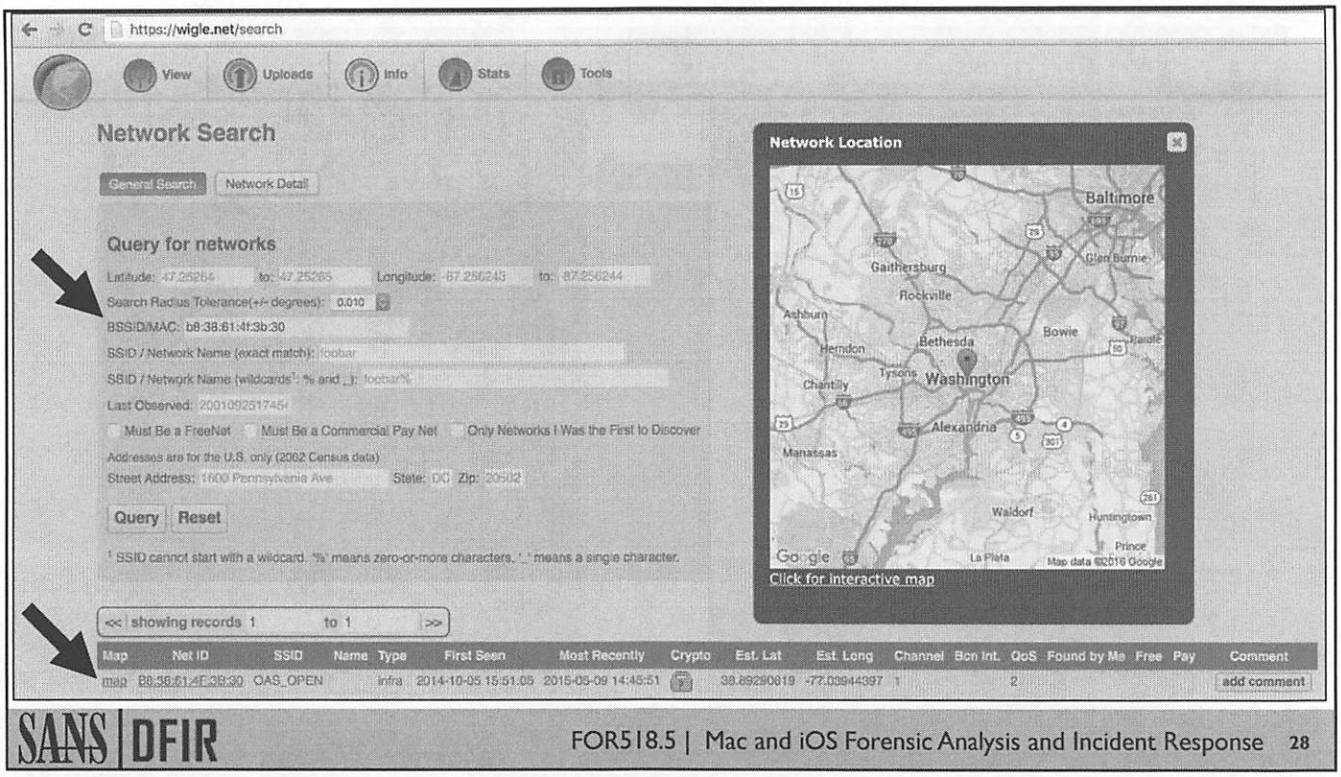
```
select
datetime (timestamp+978307200,'unixepoch','localtime') as Timestamp,
MAC,
Channel,
Latitude,
Longitude
from WifiLocation
order by Timestamp
```



iOS Physical: `/private/var/root/library/caches/locationd/cache_encryptedB.db`
macOS:
`/private/var/folders/zz/zyxvpxvq6csfxvn_n00000sm00006d/cache_encryptedA.db` or `lockCache_encryptedA.db` (Directory names may differ.)

We can change the MAC address to something that we have seen before using the Calculator application in macOS. We need to use the "`Programmer`" View to do this. If yours is in Basic mode, go to the "`View`" menu and change it to "`Programmer`" mode to be able to change bases.

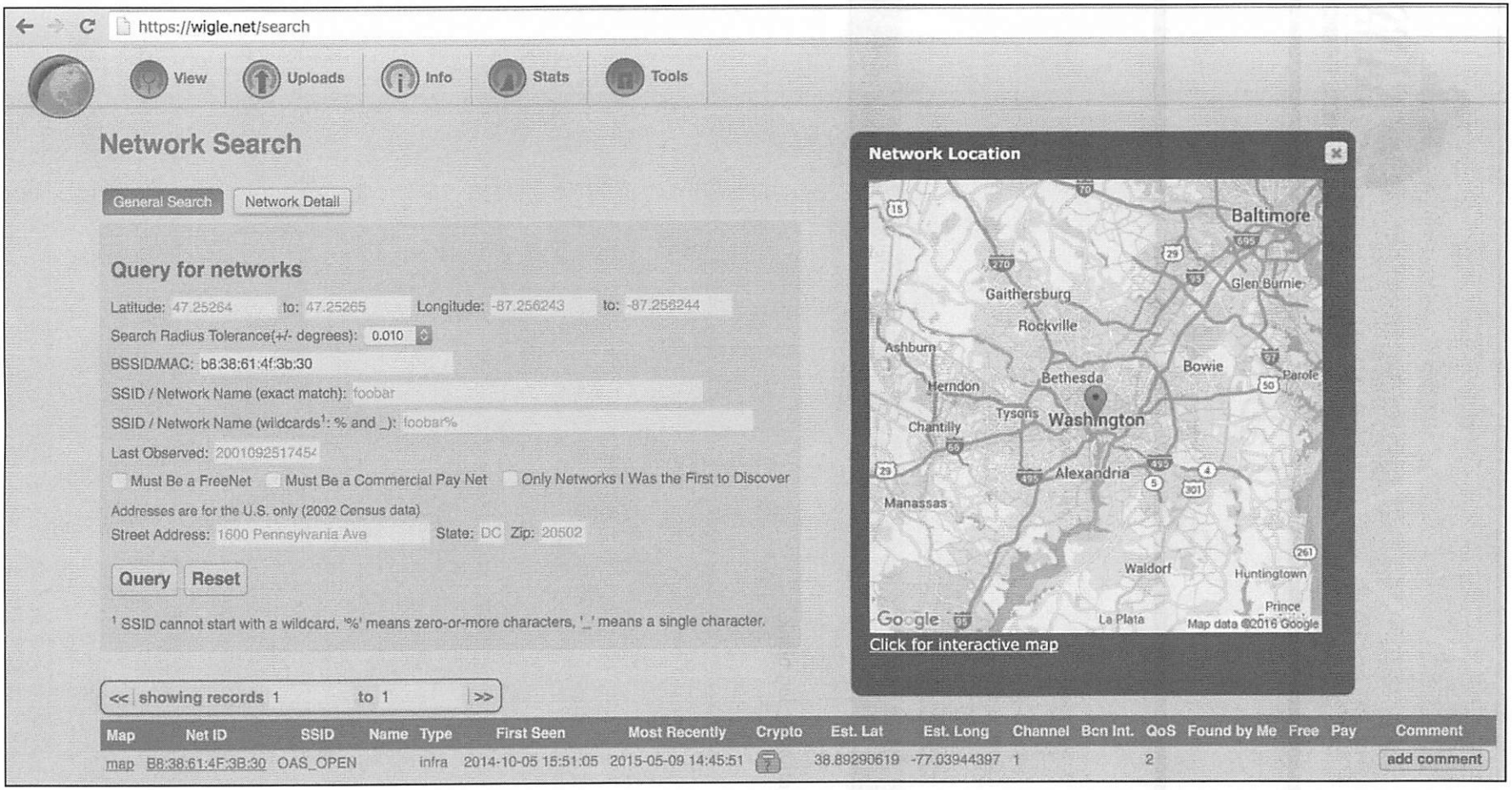
In the left screenshot, the Base10 MAC address was extracted and copied into the Calculator.app. Notice that Base10 is selected in the highlighted section. If we select the Base16 "`16`" button we can change it to hex—the format we normally see MAC addresses in (you may choose to put colons ":" between each octet).

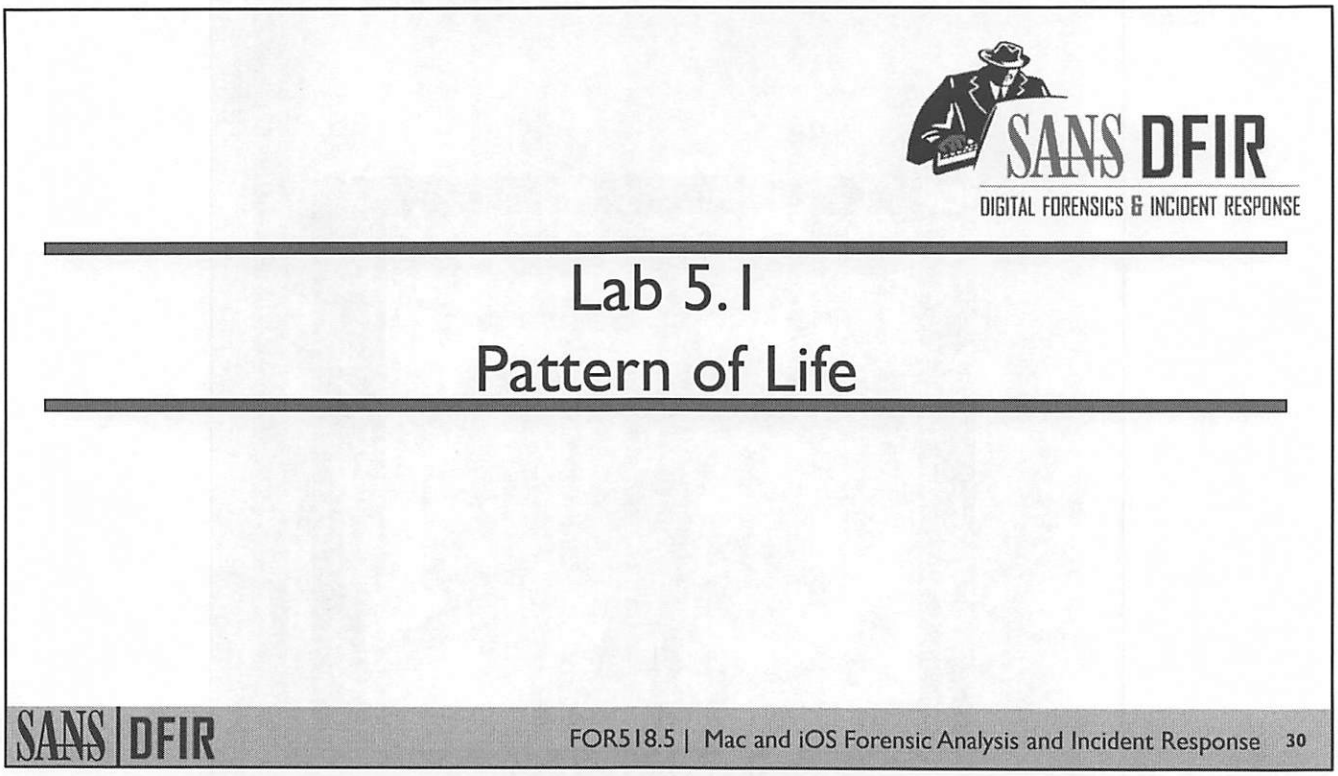


Using the hex format of the MAC address, we can now search for it on services like `wigle.net` to see what other data may be associated with it.

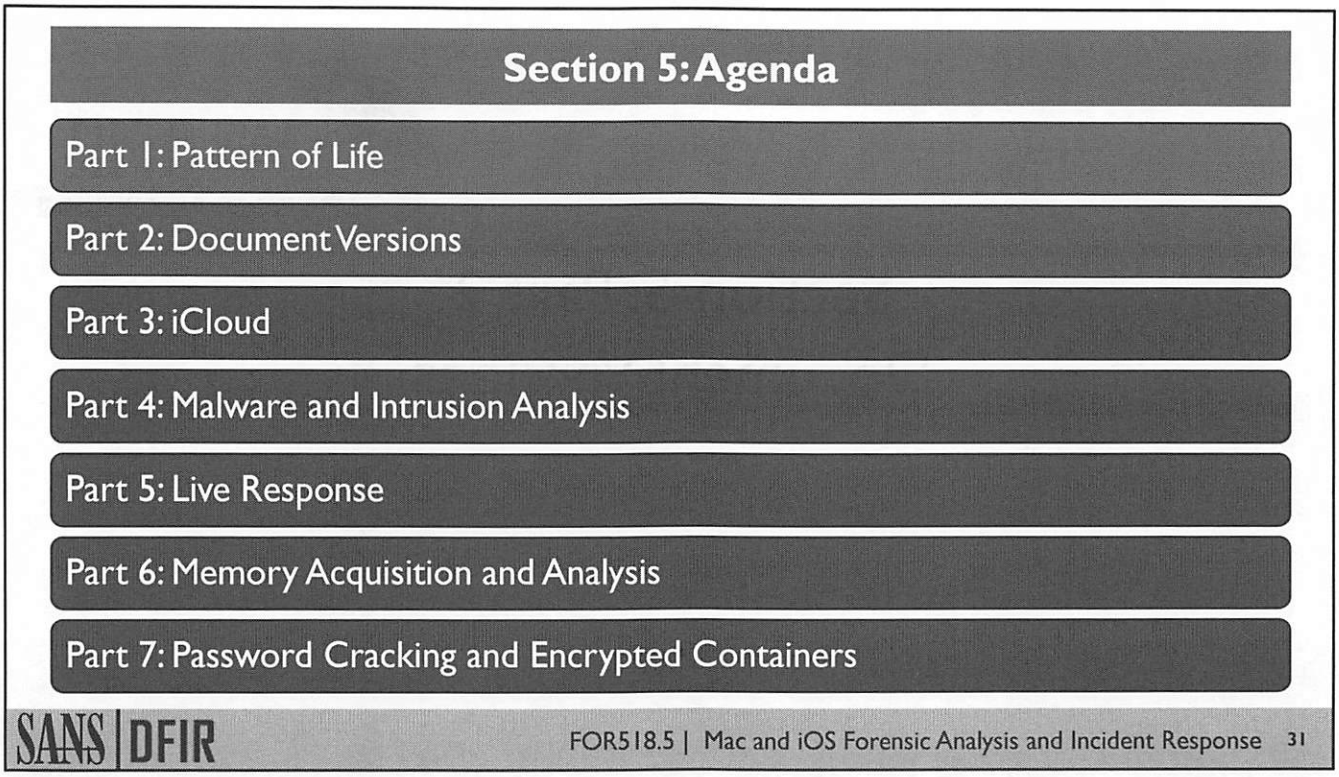
This website (shown above) collects wardriving data from a variety of sources and can be used to provide a name to an access point that has been collected, which may be of investigative value.

Reference:
`Wigle.net`

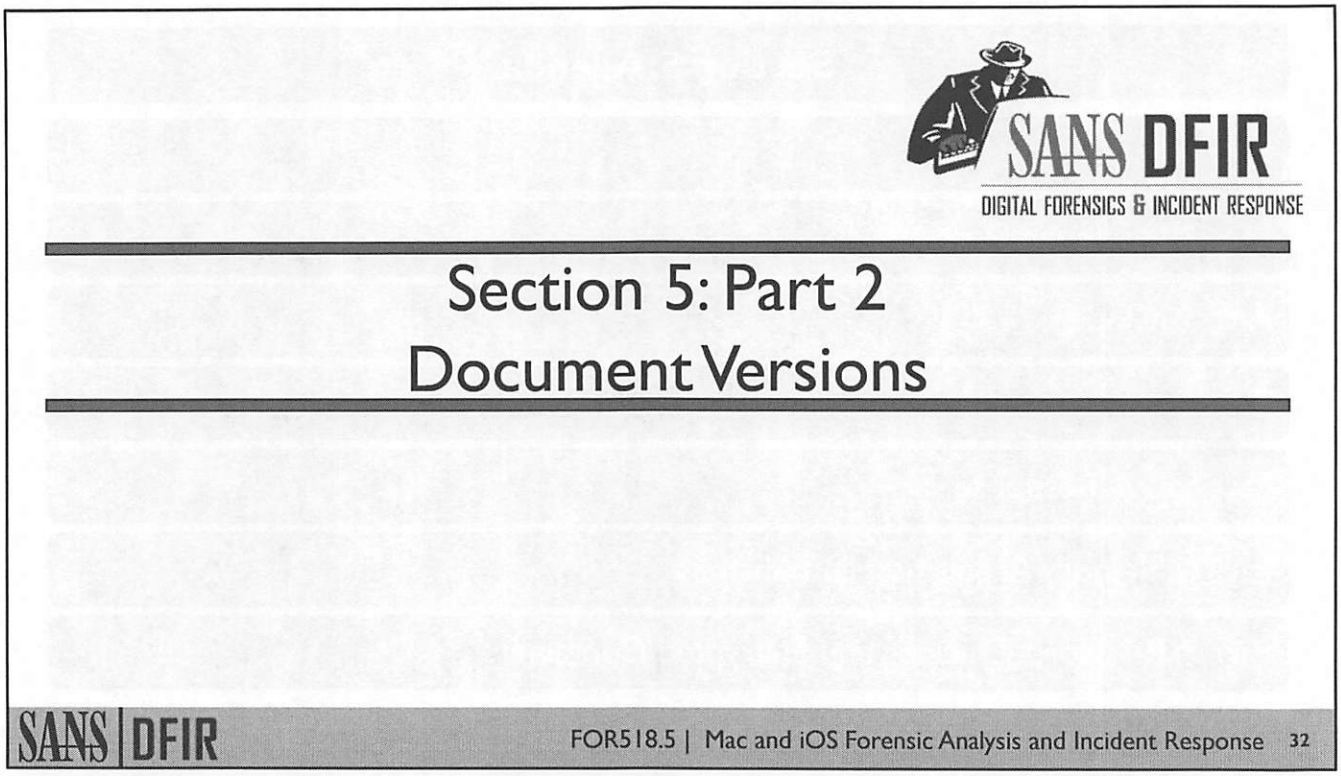




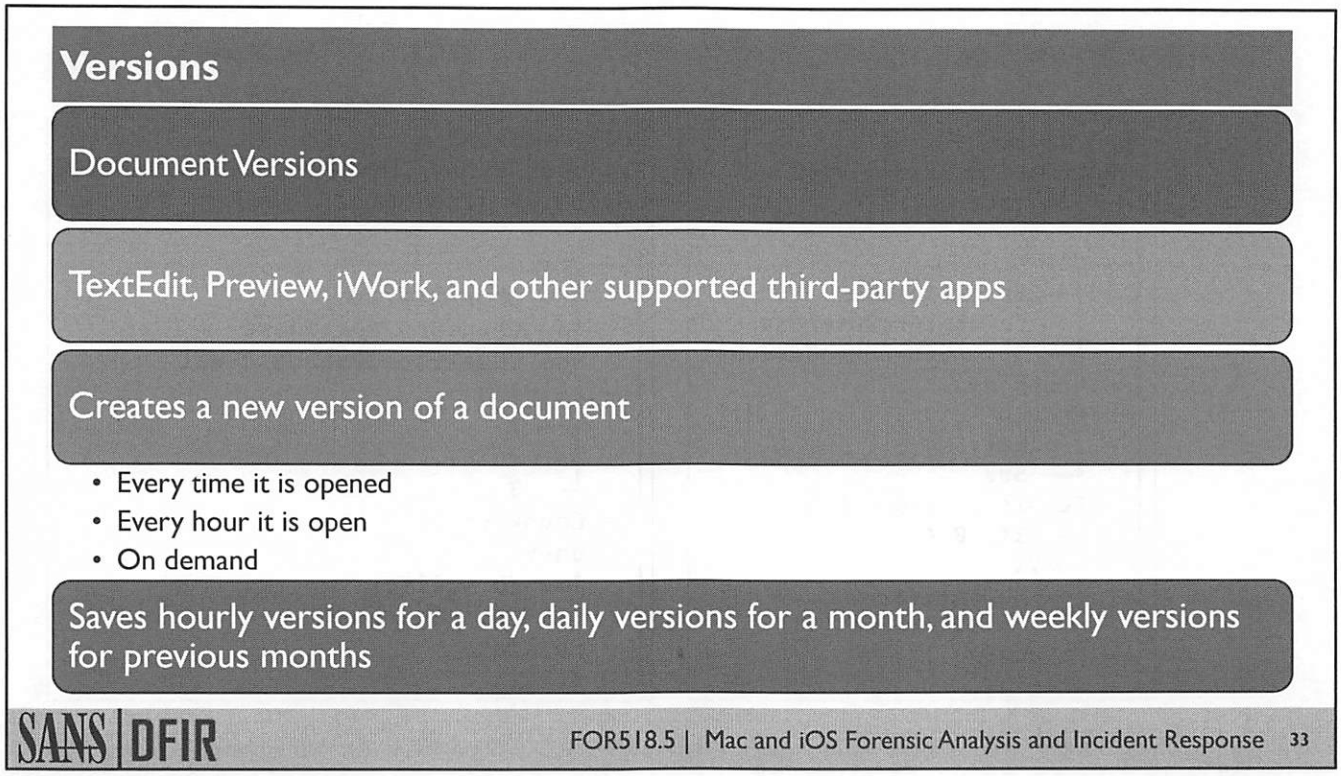
This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.



Versions was introduced in 10.7 as a way for the OS X system to automatically create document versions and back up copies of certain types of documents for the users. Ideally, this is meant as an easy way for a user to revert back to a previous version of the document, or to restore a document after a system crash.

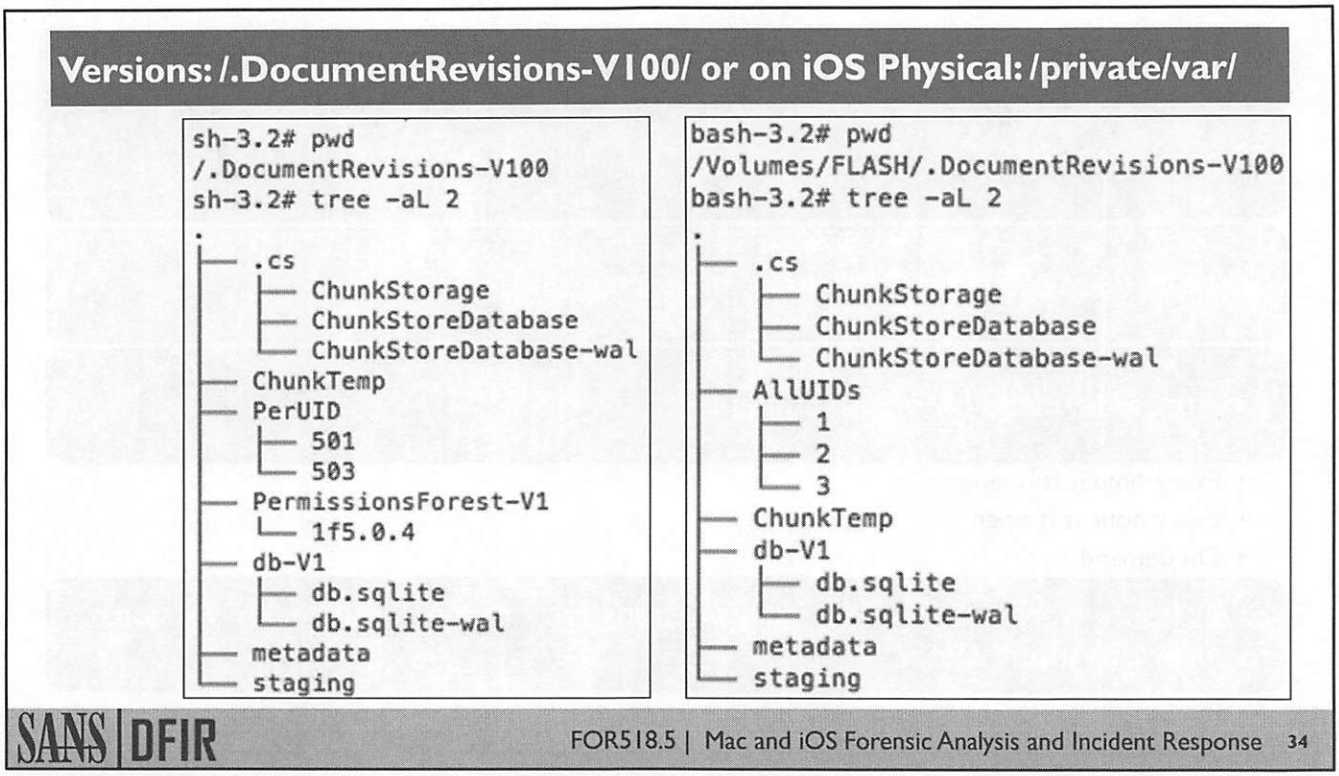
Versions will periodically create a new version of the document. For example, it will create one every time the document is opened, every hour after it is open, and when the user decides. The hourly document versions will be saved for a day, daily versions will be saved for a month, and weekly versions for the previous months.

Only certain programs on the Mac support this feature, including:

- TextEdit
- iWork applications (Numbers, Keynote, Pages)
- Preview

Other third-party apps may also support this; however, it is not mandatory. Microsoft Office does not implement it, as it uses its own version of autosave.

References:
`https://support.apple.com/en-us/HT202255`



Versions allows documents and files to have multiple "generations". Only one file for each document actually exists on the system, while changes are found in Chunk Storage (discussed later).

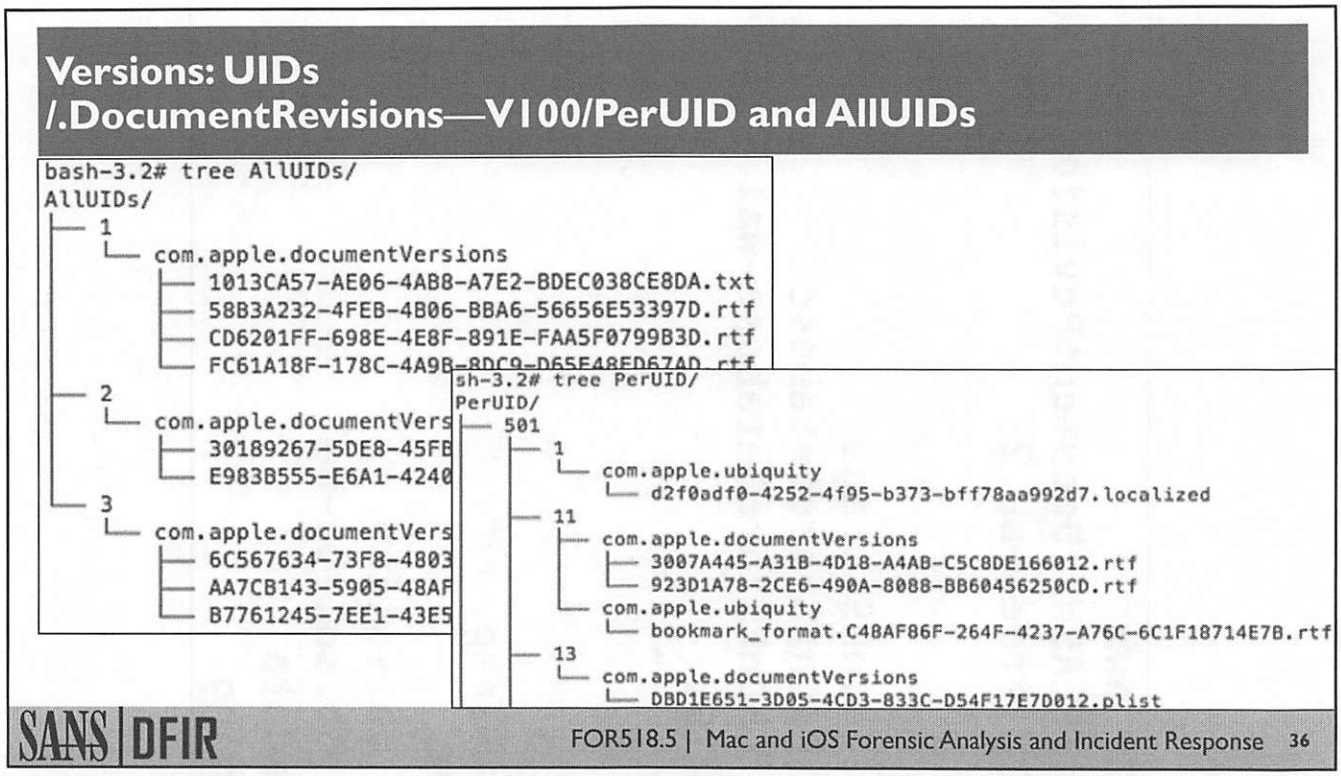
Each volume will contain a hidden directory called `.DocumentRevisions-V100`. Depending on the type of volume, the contents of this directory may be different.

The example on the left is from a system volume with a bootable version of OS X, while the example on the right is from a USB drive. The major difference between the two examples is the `PerUID`/`AllUIDs` directories (and the existence of the `PermissionsForest-V1` directory—which may or may not be present).

The system volume allows document revisions per each user under their specific user ID (501, 503, etc.), while the USB drive lumps them all under `AllUIDs`.

```
sh-3.2# pwd
/.DocumentRevisions-V100
sh-3.2# tree -aL 2
.
├── .cs
│   ├── ChunkStorage
│   ├── ChunkStoreDatabase
│   └── ChunkStoreDatabase-wal
├── ChunkTemp
├── PerUID
│   ├── 501
│   └── 503
├── PermissionsForest-V1
│   └── 1f5.0.4
├── db-V1
│   ├── db.sqlite
│   └── db.sqlite-wal
├── metadata
└── staging
```

```
bash-3.2# pwd
/Volumes/FLASH/.DocumentRevisions-V100
bash-3.2# tree -aL 2
.
├── .cs
│   ├── ChunkStorage
│   ├── ChunkStoreDatabase
│   └── ChunkStoreDatabase-wal
├── AllUIDs
│   ├── 1
│   ├── 2
│   └── 3
├── ChunkTemp
├── db-V1
│   ├── db.sqlite
│   └── db.sqlite-wal
├── metadata
└── staging
```



The top screenshot shows an example of a USB `AllUIDs` directory, while the bottom screenshot shows a system volume's `PerUID` directory (partial contents for brevity).

Under the `AllUIDs` or `PerUID/<UID>` directories, there may be a number of directories with hex filenames. These numbered directories start at 1 and count sequentially in hex numbers (i.e., 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, 11, 12, 13, 14 …). These directory names are unique across all UIDs on system volumes.

Under the hex numbered directories are other directories named in reverse DNS format:

- `com.apple.documentVersions`: Documents that are saved on the local volume
- `com.apple.ubiquity`: Documents that are saved on the local volume and iCloud

Each document "generation" is saved with a different GUID filename with the original file extension in a number directory.

Filenames under the `com.apple.ubiquity` directory will have the base filename before the GUID.

```
bash-3.2# tree AllUIDs/
AllUIDs/
├── 1
│   └── com.apple.documentVersions
│       ├── 1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt
│       ├── 58B3A232-4FEB-4B06-BBA6-56656E53397D.rtf
│       ├── CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf
│       └── FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf
├── 2
│   └── com.apple.documentVersions
│       ├── 30189267-5DE8-45FB-A240-2B75DA108ADB.txt
│       └── E983B555-E6A1-4240-A91F-D381585AB96F.txt
└── 3
    └── com.apple.documentVersions
        ├── 6C567634-73F8-4803-AD96-30876E416FEE.rtf
        ├── AA7CB143-5905-48AF-8549-D0C6BBA12542.rtf
        └── B7761245-7EE1-43E5-AC84-54E67C31DDB6.rtf
```

```
sh-3.2# tree PerUID/
PerUID/
├── 501
│   ├── 1
│   │   └── com.apple.ubiquity
│   │       └── d2f0adf0-4252-4f95-b373-bff78aa992d7.localized
│   ├── 11
│   │   ├── com.apple.documentVersions
│   │   │   ├── 3007A445-A31B-4D18-A4AB-C5C8DE166012.rtf
│   │   │   └── 923D1A78-2CE6-490A-8088-8B60456250CD.rtf
│   │   └── com.apple.ubiquity
│   │       └── bookmark_format.C48AF86F-264F-4237-A76C-6C1F18714E7B.rtf
│   └── 13
│       └── com.apple.documentVersions
│           └── DBD1E651-3D05-4CD3-833C-D54F17E7D012.plist
```

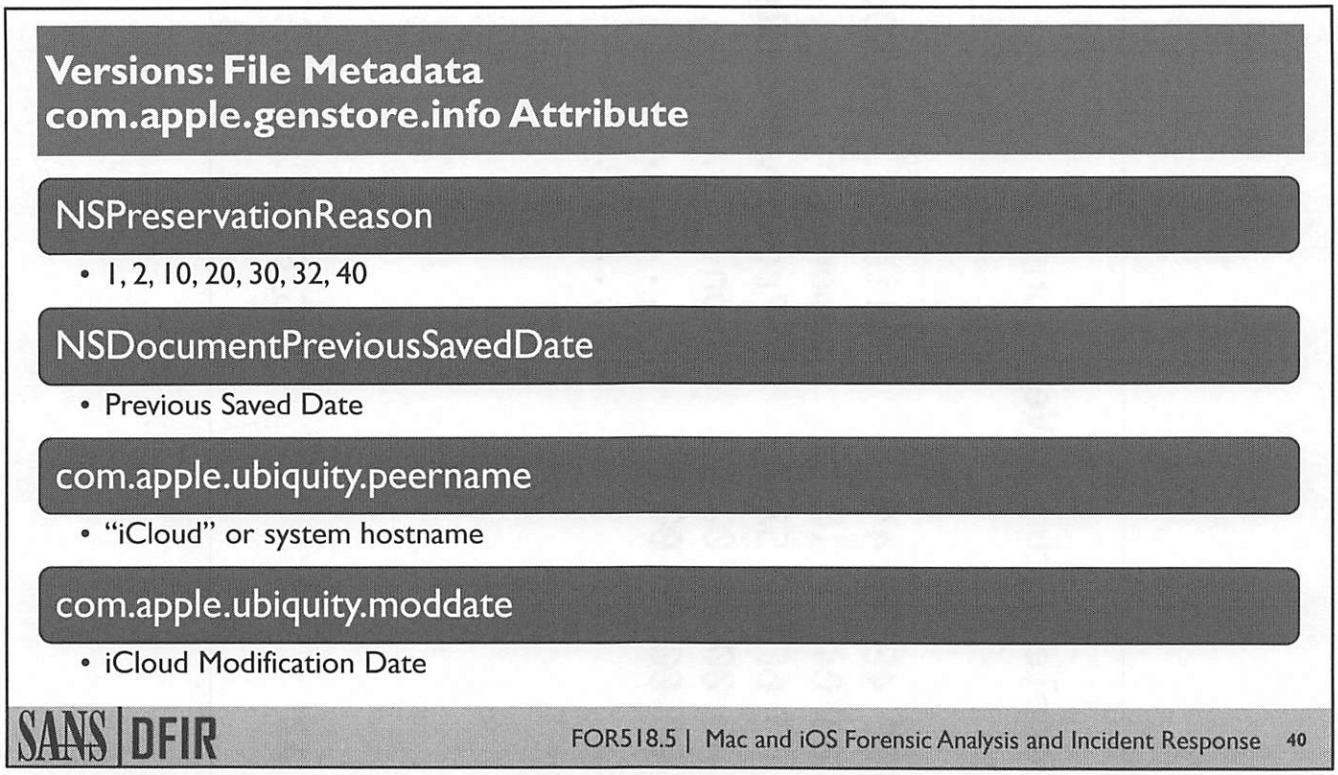
Versions: File Metadata—com.apple.genstore.*

```
bash-3.2# pwd
/.DocumentRevisions-V100/PerUID/501/2e/com.apple.documentVersions
bash-3.2# xattr -xl *
com.apple.genstore.info:
00000000  62 70 6C 69 73 74 30 30 D1 01 02 5E 4E 53 44 6F  |bplist00...^NSDo|
00000010  63 75 6D 65 6E 74 49 6E 66 6F D1 03 04 5F 10 14  |cumentInfo..._..|
00000020  4E 53 50 72 65 73 65 72 76 61 74 69 6F 6E 52 65  |NSPreservationRe|
00000030  61 73 6F 6E 10 14 08 0B 1A 1D 34 00 00 00 00 00  |ason......4.....|
00000040  00 01 01 00 00 00 00 00 00 00 05 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00 00 00 36                 |..........6|
0000005b
com.apple.genstore.orig_perms_v1:
00000000  04                                               |.|
00000001
com.apple.genstore.origdisplayname:
00000000  49 4D 47 5F 30 31 39 30 2E 6A 70 67              |IMG_0190.jpg|
0000000c
```


Each generation file has extended attributes associated with "genstore", or generational storage.

- `com.apple.genstore.origdisplayname`: Filename associated with this generation of the file.
- `com.apple.genstore.orig_perms_v1`: Examples that I've seen only have 0x04 or 0x1C as its contents.
- `com.apple.genstore.info`: Embedded binary property list file.

```
bash-3.2# pwd
/.DocumentRevisions-V100/PerUID/501/2e/com.apple.documentVersions
bash-3.2# xattr -xl *
com.apple.genstore.info:
00000000  62 70 6C 69 73 74 30 30 D1 01 02 5E 4E 53 44 6F  |bplist00...^NSDo|
00000010  63 75 6D 65 6E 74 49 6E 66 6F D1 03 04 5F 10 14  |cumentInfo..._..|
00000020  4E 53 50 72 65 73 65 72 76 61 74 69 6F 6E 52 65  |NSPreservationRe|
00000030  61 73 6F 6E 10 14 08 0B 1A 1D 34 00 00 00 00 00  |ason......4.....|
00000040  00 01 01 00 00 00 00 00 00 00 05 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00 00 00 36                 |..........6|
0000005b
com.apple.genstore.orig_perms_v1:
00000000  04                                               |.|
00000001
com.apple.genstore.origdisplayname:
00000000  49 4D 47 5F 30 31 39 30 2E 6A 70 67              |IMG_0190.jpg|
0000000c
```



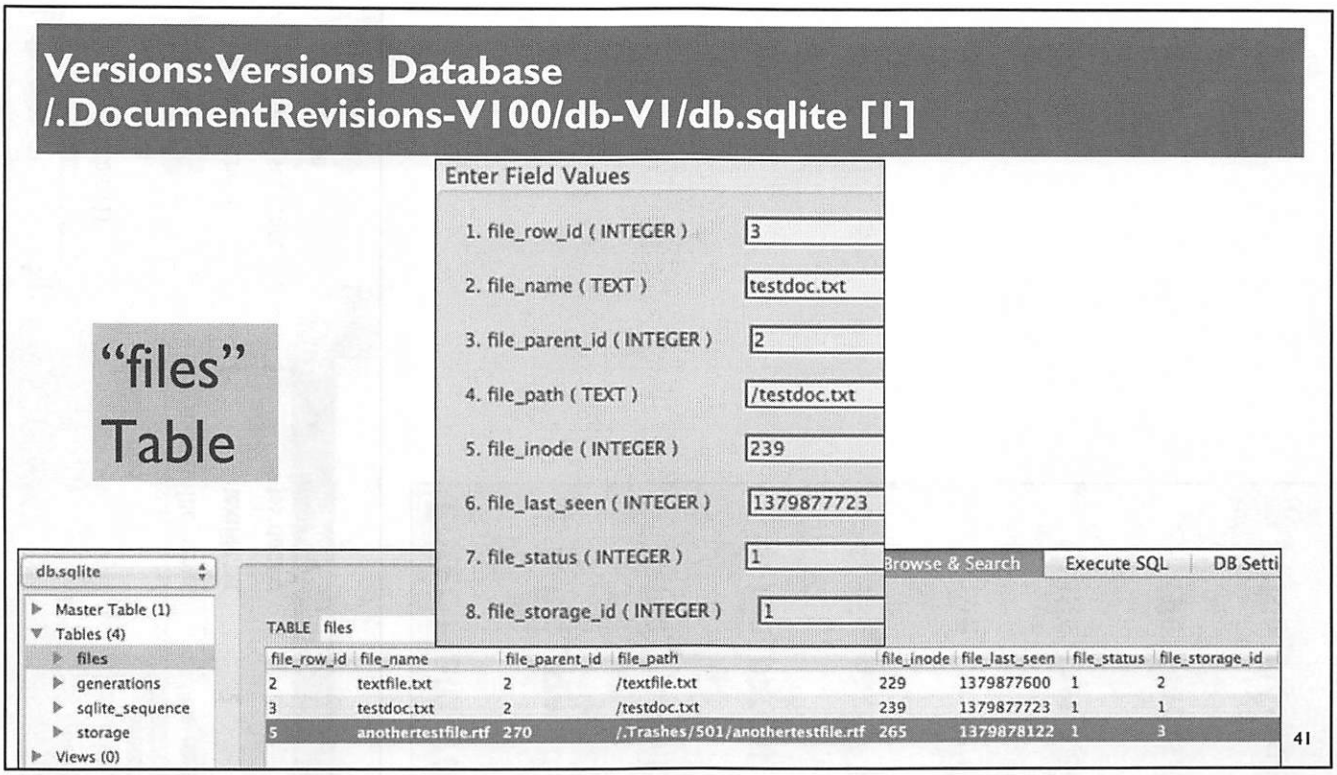
Using the command below, we can extract this binary plist (make sure you keep those dashes in there):

```
xattr -p com.apple.genstore.info <filename> | xxd -r -p | plutil -convert
xml1 - -o -
```

Once the binary property list is extracted from the `com.apple.genstore.info` attribute, we can gather more metadata about the document "generation". This property list contains the keys listed above in the slide.

The `NSPreservationReason` has a number associated with why this "generation" was created. More testing and research will have to be performed to distinguish what action is required for each of these identifiers. The `NSDocumentPreviousSavedDate` contains the date the generation was previously saved.

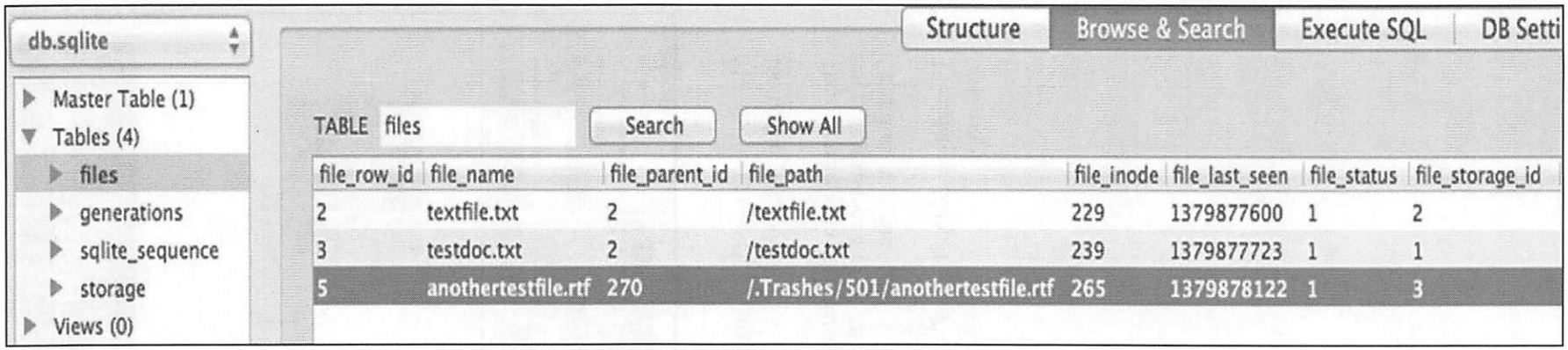
Two iCloud-related keys may be stored showing when a document was last modified (`com.apple.ubiquity.moddate`) on a specific iCloud "peer" (`com.apple.ubiquity.peername`).

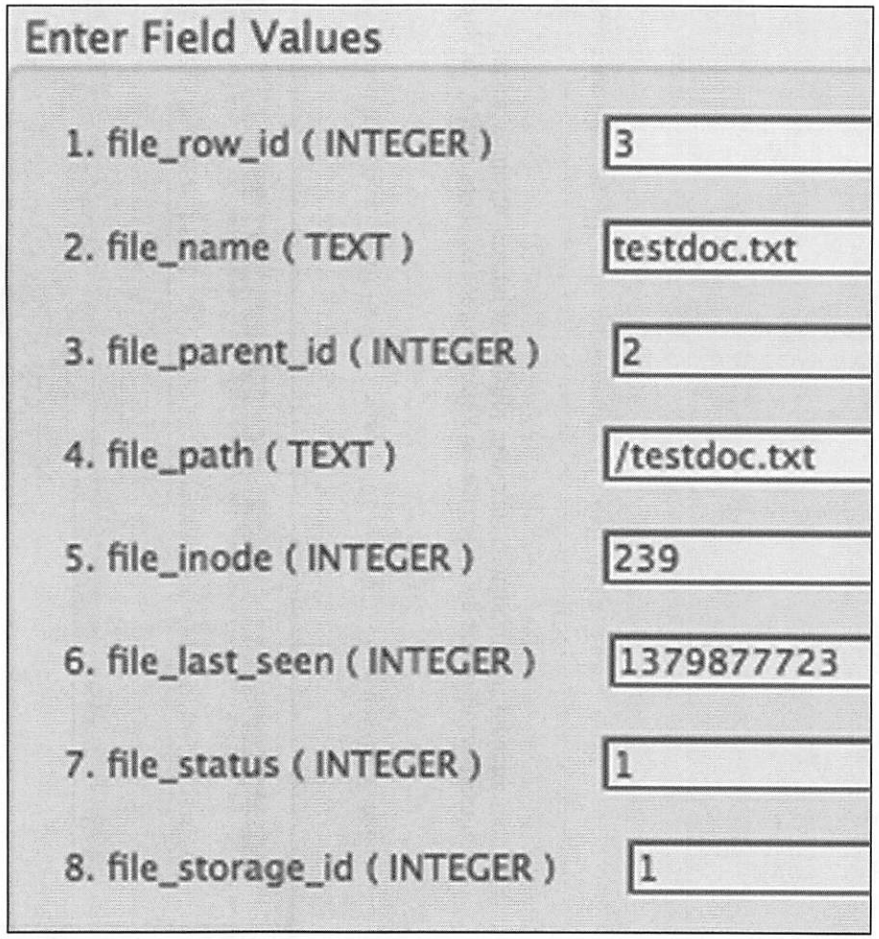


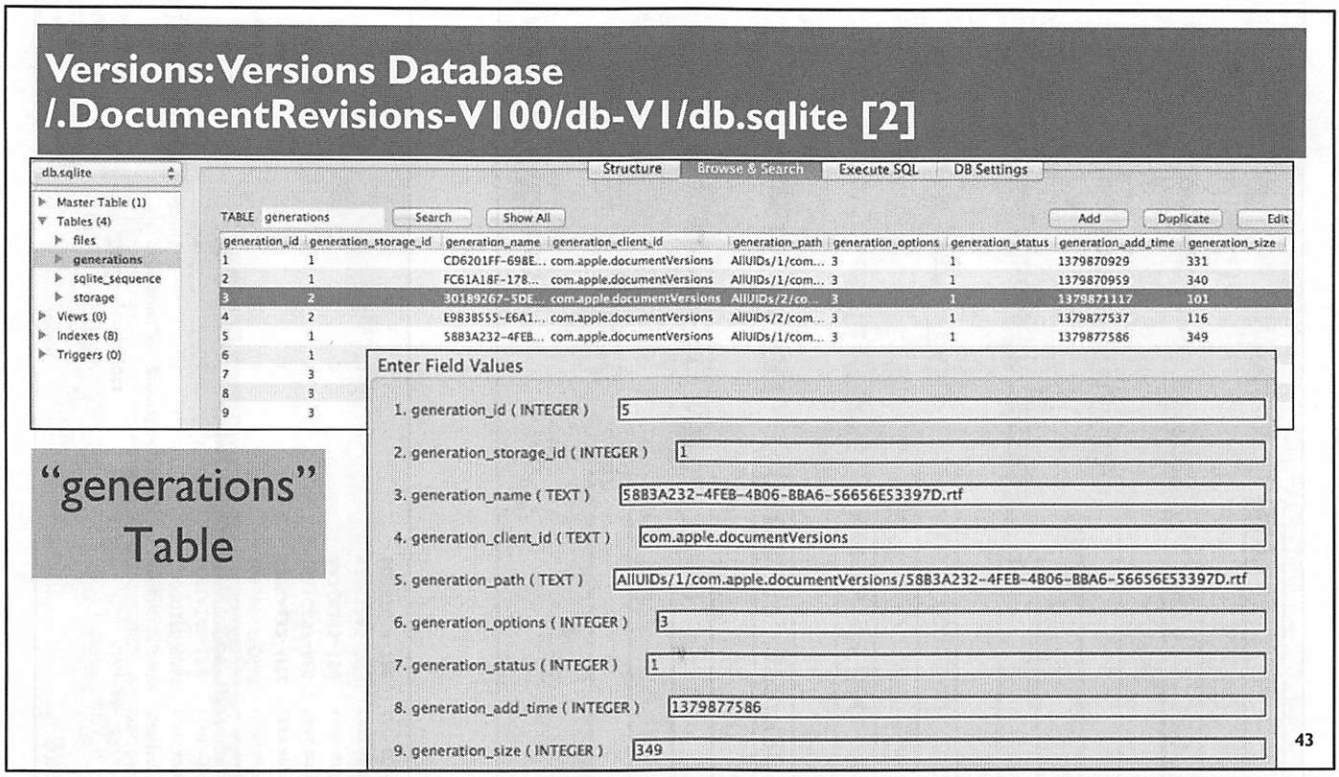
The `db.sqlite` SQLite database located in `/.DocumentRevisions-V100/db-V1/` contains the metadata for the file "versions".

These screenshots were created using the SQLite Manager for the Firefox browser. The top screenshot shows the larger database view of the "`files`" tables, while the bottom screenshot shows the contents of one tuple for the `testdoc.txt` file.

The database table, "`files`" contains the filename, file path, inode number (CNID), and a "`last seen`" timestamp.

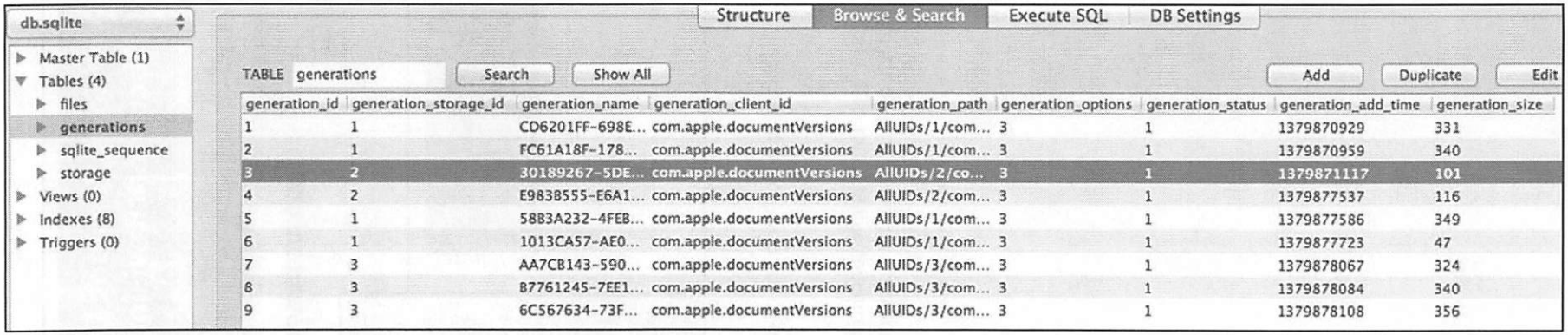






The database table, "`generations`" in the `db.sqlite` database, contains the details about each document "generation".

Each "generation" tuple contains the generation GUID, the path to the "generation", a timestamp when it was added, and the size of the "generation".



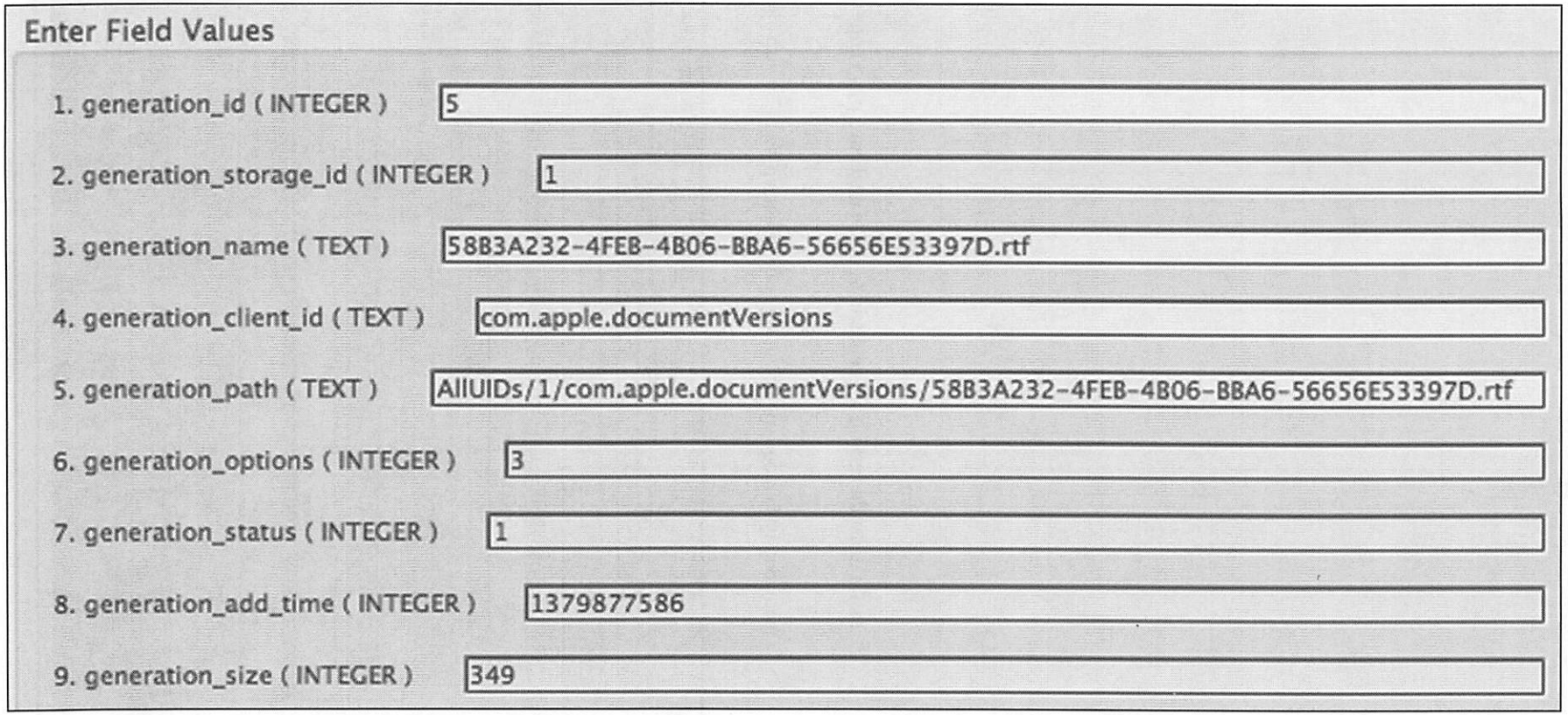
db.sqlite

- Master Table (1)
- Tables (4)
  - files
  - generations
  - sqlite_sequence
  - storage
- Views (0)
- Indexes (8)
- Triggers (0)

Structure | Browse & Search | Execute SQL | DB Settings

TABLE generations | Search | Show All | Add | Duplicate | Edit

generation_id	generation_storage_id	generation_name	generation_client_id	generation_path	generation_options	generation_status	generation_add_time	generation_size
1	1	CD6201FF-698E...	com.apple.documentVersions	AllUIDs/1/com...	3	1	1379870929	331
2	1	FC61A18F-178...	com.apple.documentVersions	AllUIDs/1/com...	3	1	1379870959	340
3	2	30189267-5DE...	com.apple.documentVersions	AllUIDs/2/co...	3	1	1379871117	101
4	2	E983B555-E6A1...	com.apple.documentVersions	AllUIDs/2/com...	3	1	1379877537	116
5	1	58B3A232-4FEB...	com.apple.documentVersions	AllUIDs/1/com...	3	1	1379877586	349
6	1	1013CA57-AE0...	com.apple.documentVersions	AllUIDs/1/com...	3	1	1379877723	47
7	3	AA7CB143-590...	com.apple.documentVersions	AllUIDs/3/com...	3	1	1379878067	324
8	3	B7761245-7EE1...	com.apple.documentVersions	AllUIDs/3/com...	3	1	1379878084	340
9	3	6C567634-73F...	com.apple.documentVersions	AllUIDs/3/com...	3	1	1379878108	356



Enter Field Values

1. generation_id (INTEGER) 5
2. generation_storage_id (INTEGER) 1
3. generation_name (TEXT) 58B3A232-4FEB-4B06-BBA6-56656E53397D.rtf
4. generation_client_id (TEXT) com.apple.documentVersions
5. generation_path (TEXT) AllUIDs/1/com.apple.documentVersions/58B3A232-4FEB-4B06-BBA6-56656E53397D.rtf
6. generation_options (INTEGER) 3
7. generation_status (INTEGER) 1
8. generation_add_time (INTEGER) 1379877586
9. generation_size (INTEGER) 349

Versions: "Generation" Files
./DocumentRevisions/*UIDs/#/com.apple.documentVersions

```
bash-3.2# pwd
/Volumes/FLASH/.DocumentRevisions-V100/AllUIDs/1/com.apple.documentVersions
bash-3.2# ls -l
total 0
-r--r--r--@ 1 _unknown  _unknown   47 Sep 22 15:20 1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt
-r--r--r--@ 1 _unknown  _unknown  349 Sep 22 13:29 58B3A232-4FEB-4B06-BBA6-56656E53397D.rtf
-r--r--r--@ 1 _unknown  _unknown  331 Sep 22 13:28 CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf
-r--r--r--@ 1 _unknown  _unknown  340 Sep 22 13:28 FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf
```

Name	Date Created	Date Modified	Date Accessed	Date Added	Size
FLASH	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)		--
.apdisk	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	305 Bytes
.DocumentRevisions-V100	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
.cs	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
AllUIDs	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
1	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
com.apple.documentVersions	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
5883A232-4FEB-4B06-B8A6-56656E53397D.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes

In the top screenshot, the output of an `ls` command in Terminal shows the document "generations" for one file.

The fifth column in this Terminal output is the expected file size for each "generation" of the file. It may look like these files are all individual files, but the OS X system is only supposed to store one copy of each document, right?

The bottom screenshot shows the same files in BlackLight. Note the sizes are 0 bytes… so where is the data actually stored? … ChunkStorage!

On a side note, notice how the file extension for these files changes from `.rtf` to `.txt` over a period of time. This was caused by the user selecting "`Make Plaintext`" from the Format menu in TextEdit.

```
bash-3.2# pwd
/Volumes/FLASH/.DocumentRevisions-V100/AllUIDs/1/com.apple.documentVersions
bash-3.2# ls -l
total 0
-r--r--r--@ 1 _unknown  _unknown   47 Sep 22 15:20 1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt
-r--r--r--@ 1 _unknown  _unknown  349 Sep 22 13:29 58B3A232-4FEB-4B06-BBA6-56656E53397D.rtf
-r--r--r--@ 1 _unknown  _unknown  331 Sep 22 13:28 CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf
-r--r--r--@ 1 _unknown  _unknown  340 Sep 22 13:28 FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf
```

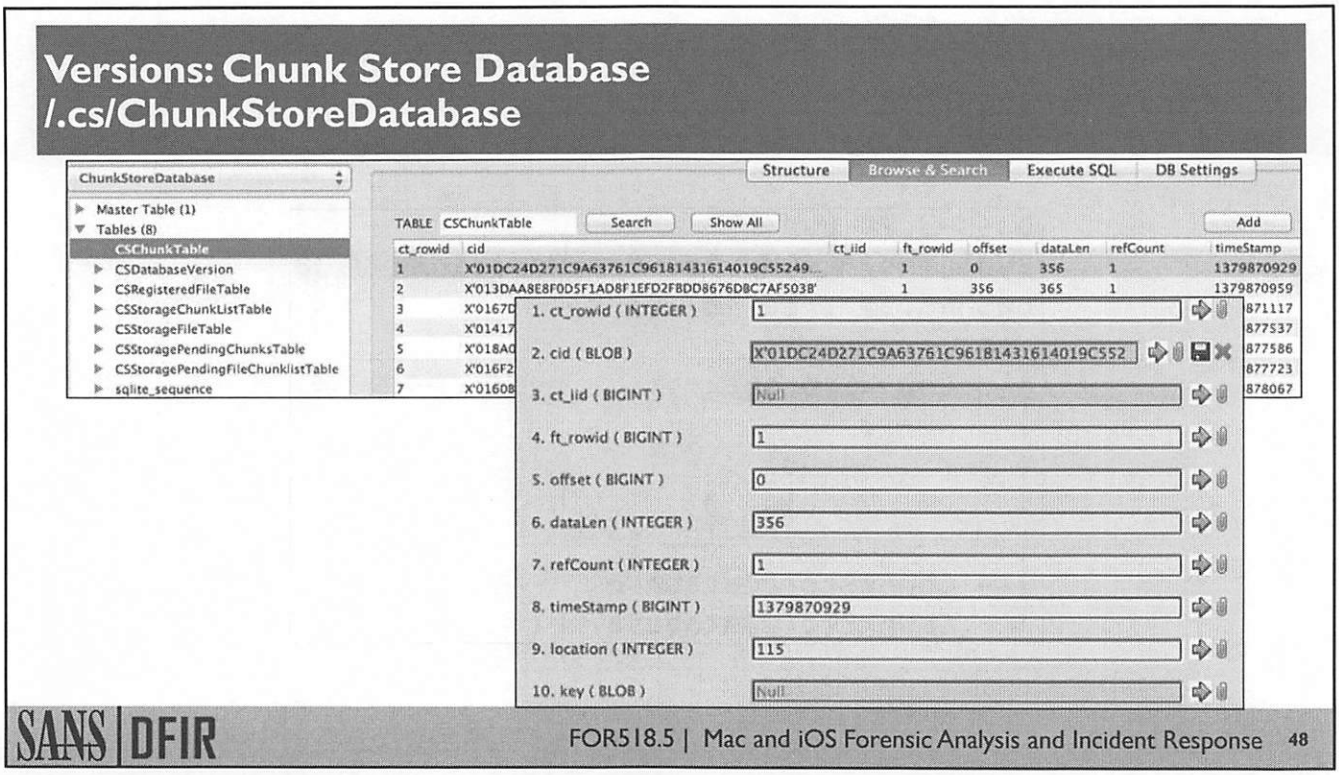
Name	Date Created	Date Modified	Date Accessed	Date Added	Size
▼ FLASH	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)		--
.apdisk	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	305 Bytes
▼ .DocumentRevisions-V100	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
▶ .cs	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
▼ AllUIDs	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
▼ 1	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
▼ com.apple.documentVersions	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	--
1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
5883A232-4FEB-4B06-BBA6-56656E53397D.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes
FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	2013-09-22 (UTC)	0 Bytes

Versions: Chunk Storage /.DocumentRevisions-V100/.cs/

```
bash-3.2# pwd
/Volumes/FLASH/.DocumentRevisions-V100
bash-3.2# tree -a .cs
.cs
├── ChunkStorage
│   └── 0
│       └── 0
│           └── 0
│               └── 1
├── ChunkStoreDatabase
└── ChunkStoreDatabase-wal
```


The data is saved in something called Chunk Storage. All the expected data supposedly stored in the "generations" files is stored in a file as "chunks" that is managed by a SQLite database.

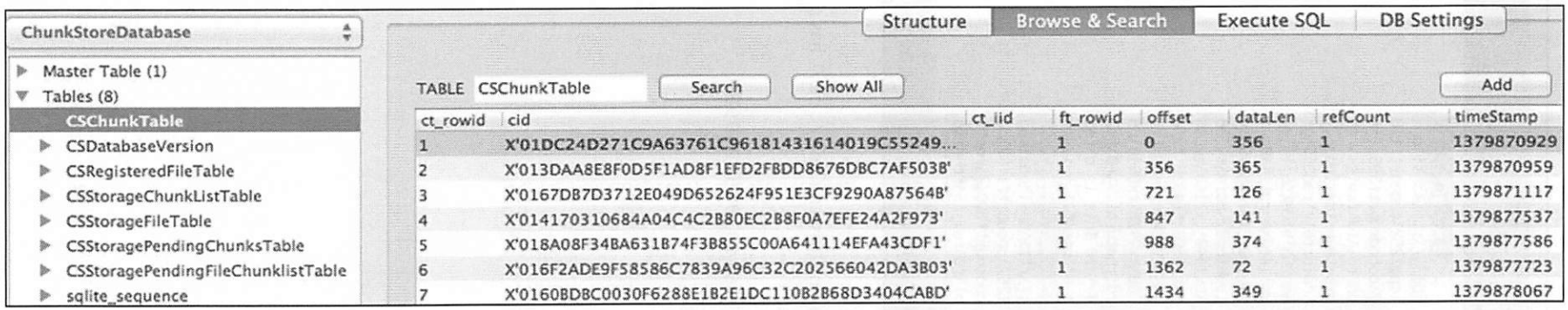
The screenshot above shows a tree output showing the structure of the hidden `.cs` directory located in the `/.DocumentRevisions-V100/` directory. The Chunk Storage files are saved in the nested subdirectories under the `ChunkStorage` directory. The SQLite database controlling this storage is the `ChunkStoreDatabase` file.

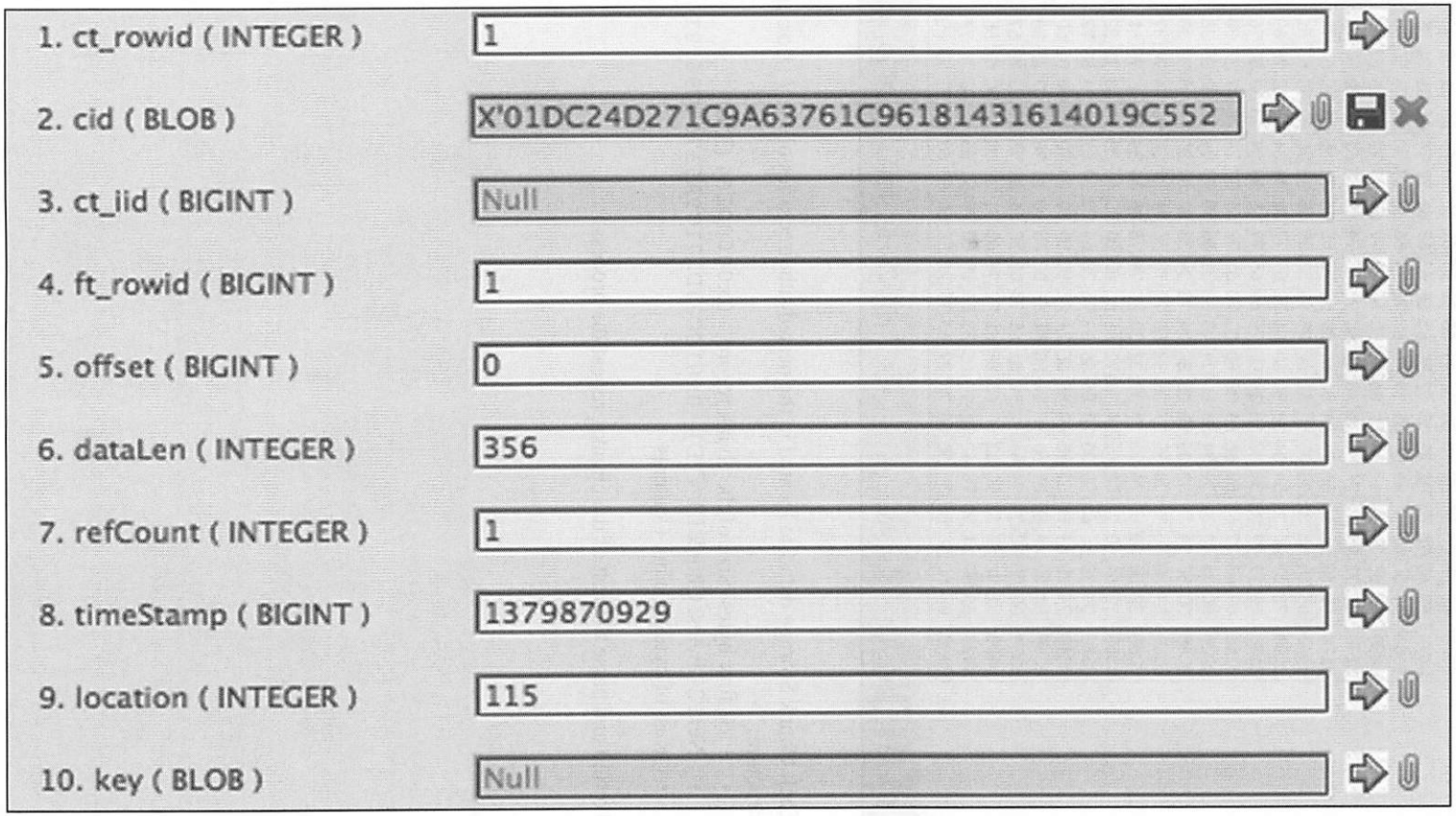


The SQLite database, `ChunkStoreDatabase`, contains the information needed to get at the stored "chunks" and to reassemble them as needed.

The `CSChunkTable`, shown in the top screenshot, contains a row for each "chunk". Each "chunk" is defined by the following:

- `cid`: Chunk ID
- `offset`: Offset in the ChunkStorage data file
- `dataLen`: A data length
- `timeStamp`: When this "chunk" was stored





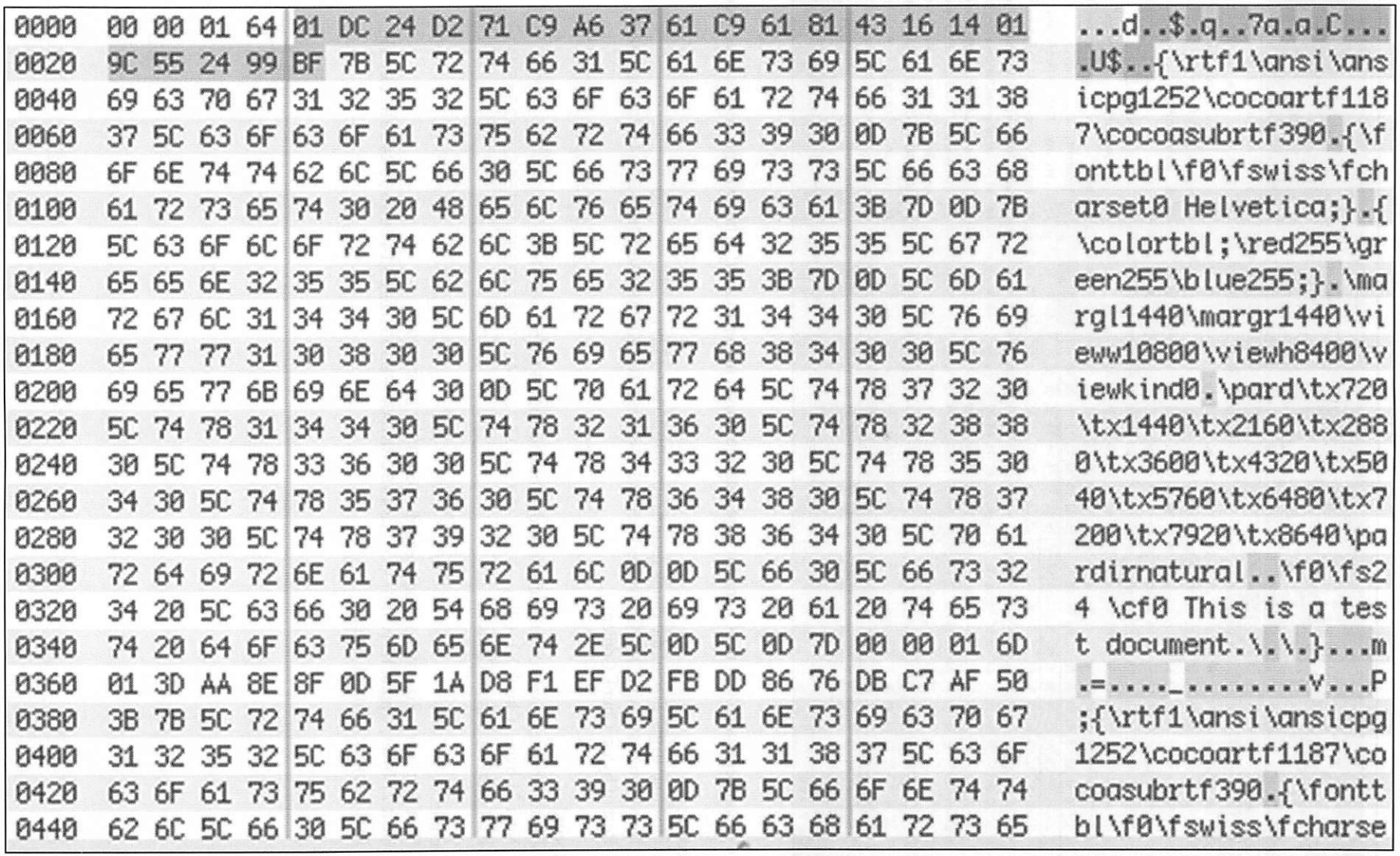
Versions: Chunk Storage—Chunk Storage Files

```
0000  00 00 01 64 01 DC 24 D2 71 C9 A6 37 61 C9 61 81 43 16 14 01   ...d..$.q..7a.a.C...
0020  9C 55 24 99 BF 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73   .U$..{\rtf1\ansi\ans
0040  69 63 70 67 31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38   icpg1252\cocoartf118
0060  37 5C 63 6F 63 6F 61 73 75 62 72 74 66 33 39 30 0D 7B 5C 66   7\cocoasubrtf390.{\f
0080  6F 6E 74 74 62 6C 5C 66 30 5C 66 73 77 69 73 73 5C 66 63 68   onttbl\f0\fswiss\fch
0100  61 72 73 65 74 30 20 48 65 6C 76 65 74 69 63 61 3B 7D 0D 7B   arset0 Helvetica;}.{
0120  5C 63 6F 6C 6F 72 74 62 6C 3B 5C 72 65 64 32 35 35 5C 67 72   \colortbl;\red255\gr
0140  65 65 6E 32 35 35 5C 62 6C 75 65 32 35 35 3B 7D 0D 5C 6D 61   een255\blue255;}.\ma
0160  72 67 6C 31 34 34 30 5C 6D 61 72 67 72 31 34 34 30 5C 76 69   rgl1440\margr1440\vi
0180  65 77 77 31 30 38 30 30 5C 76 69 65 77 68 38 34 30 30 5C 76   eww10800\viewh8400\v
0200  69 65 77 6B 69 6E 64 30 0D 5C 70 61 72 64 5C 74 78 37 32 30   iewkind0.\pard\tx720
0220  5C 74 78 31 34 34 30 5C 74 78 32 31 36 30 5C 74 78 32 38 38   \tx1440\tx2160\tx288
0240  30 5C 74 78 33 36 30 30 5C 74 78 34 33 32 30 5C 74 78 35 30   0\tx3600\tx4320\tx50
0260  34 30 5C 74 78 35 37 36 30 5C 74 78 36 34 38 30 5C 74 78 37   40\tx5760\tx6480\tx7
0280  32 30 30 5C 74 78 37 39 32 30 5C 74 78 38 36 34 30 5C 70 61   200\tx7920\tx8640\pa
0300  72 64 69 72 6E 61 74 75 72 61 6C 0D 0D 5C 66 30 5C 66 73 32   rdirnatural..\f0\fs2
0320  34 20 5C 63 66 30 20 54 68 69 73 20 69 73 20 61 20 74 65 73   4 \cf0 This is a tes
0340  74 20 64 6F 63 75 6D 65 6E 74 2E 5C 0D 5C 0D 7D 00 00 01 6D   t document.\.\.}...m
0360  01 3D AA 8E 8F 0D 5F 1A D8 F1 EF D2 FB DD 86 76 DB C7 AF 50   .=...._........v...P
0380  3B 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73 69 63 70 67   ;{\rtf1\ansi\ansicpg
0400  31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38 37 5C 63 6F   1252\cocoartf1187\co
0420  63 6F 61 73 75 62 72 74 66 33 39 30 0D 7B 5C 66 6F 6E 74 74   coasubrtf390.{\fontt
0440  62 6C 5C 66 30 5C 66 73 77 69 73 73 5C 66 63 68 61 72 73 65   bl\f0\fswiss\fcharse
```

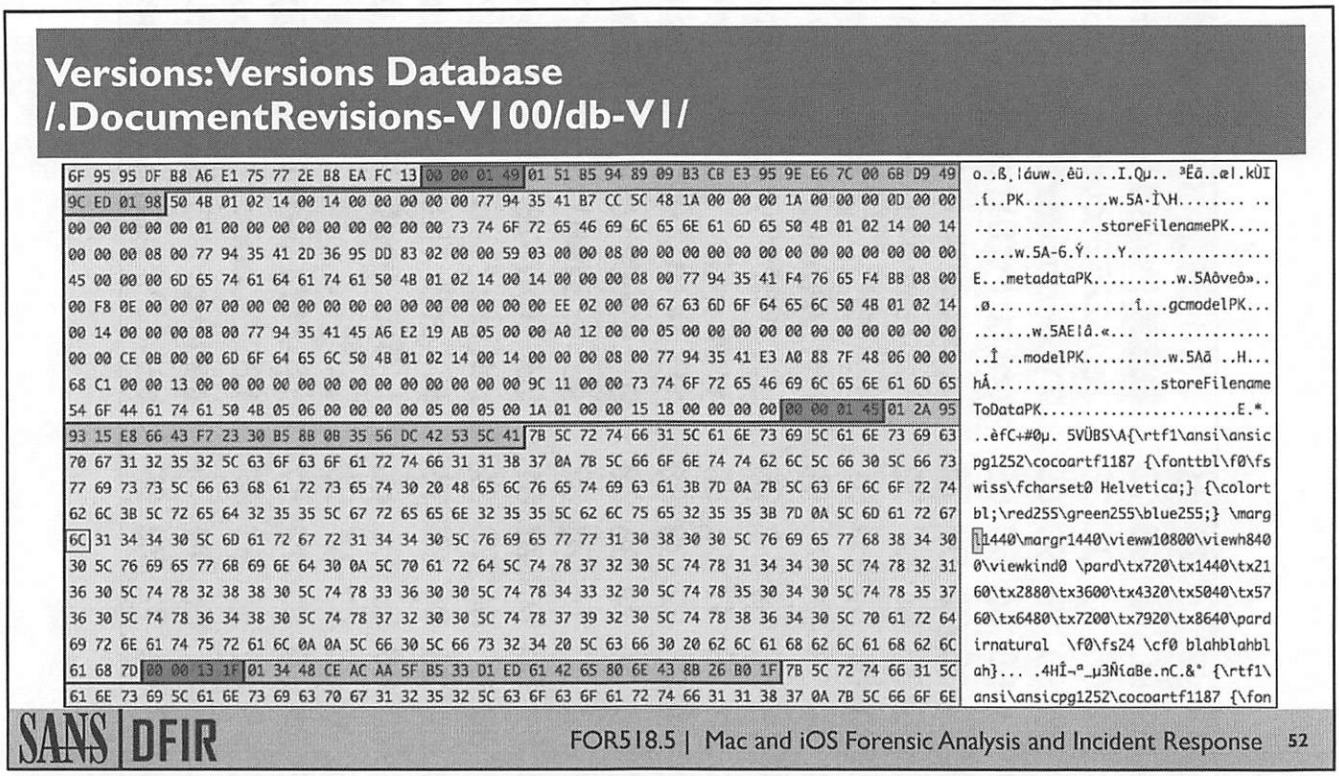

Each Chunk Storage data file located in the nested subdirectories of the `/.DocumentRevisions/.cs/` has a structure:

- First four bytes: Size of the chunk record—`0x00000164` (356)
- Next 21 bytes: Chunk ID (CID)—`0x01DC24D271C9A63761C96181431614019C552499BF`
- Rest of bytes: Chunk Contents

To find and parse the rest of the chunks, repeat the process.



```
0000  00 00 01 64 01 DC 24 D2 71 C9 A6 37 61 C9 61 81 43 16 14 01  ...d..$.q..7a.a.C...
0020  9C 55 24 99 BF 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73  .U$..{\rtf1\ansi\ans
0040  69 63 70 67 31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38  icpg1252\cocoartf118
0060  37 5C 63 6F 63 6F 61 73 75 62 72 74 66 33 39 30 0D 7B 5C 66  7\cocoasubrtf390.{\f
0080  6F 6E 74 74 62 6C 5C 66 30 5C 66 73 77 69 73 73 5C 66 63 68  onttbl\f0\fswiss\fch
0100  61 72 73 65 74 30 20 48 65 6C 76 65 74 69 63 61 3B 7D 0D 7B  arset0 Helvetica;}.{
0120  5C 63 6F 6C 6F 72 74 62 6C 3B 5C 72 65 64 32 35 35 5C 67 72  \colortbl;\red255\gr
0140  65 65 6E 32 35 35 5C 62 6C 75 65 32 35 35 3B 7D 0D 5C 6D 61  een255\blue255;}.\ma
0160  72 67 6C 31 34 34 30 5C 6D 61 72 67 72 31 34 34 30 5C 76 69  rgl1440\margr1440\vi
0180  65 77 77 31 30 38 30 30 5C 76 69 65 77 68 38 34 30 30 5C 76  eww10800\viewh8400\v
0200  69 65 77 6B 69 6E 64 30 0D 5C 70 61 72 64 5C 74 78 37 32 30  iewkind0.\pard\tx720
0220  5C 74 78 31 34 34 30 5C 74 78 32 31 36 30 5C 74 78 32 38 38  \tx1440\tx2160\tx288
0240  30 5C 74 78 33 36 30 30 5C 74 78 34 33 32 30 5C 74 78 35 30  0\tx3600\tx4320\tx50
0260  34 30 5C 74 78 35 37 36 30 5C 74 78 36 34 38 30 5C 74 78 37  40\tx5760\tx6480\tx7
0280  32 30 30 5C 74 78 37 39 32 30 5C 74 78 38 36 34 30 5C 70 61  200\tx7920\tx8640\pa
0300  72 64 69 72 6E 61 74 75 72 61 6C 0D 0D 5C 66 30 5C 66 73 32  rdirnatural..\f0\fs2
0320  34 20 5C 63 66 30 20 54 68 69 73 20 69 73 20 61 20 74 65 73  4 \cf0 This is a tes
0340  74 20 64 6F 63 75 6D 65 6E 74 2E 5C 0D 5C 0D 7D 00 00 01 6D  t document.\.\.}...m
0360  01 3D AA 8E 8F 0D 5F 1A D8 F1 EF D2 FB DD 86 76 DB C7 AF 50  .=...._........v...P
0380  3B 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73 69 63 70 67  ;{\rtf1\ansi\ansicpg
0400  31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38 37 5C 63 6F  1252\cocoartf1187\co
0420  63 6F 61 73 75 62 72 74 66 33 39 30 0D 7B 5C 66 6F 6E 74 74  coasubrtf390.{\fontt
0440  62 6C 5C 66 30 5C 66 73 77 69 73 73 5C 66 63 68 61 72 73 65  bl\f0\fswiss\fcharse
```



Above is another example of the Chunk Storage data file shown in Synalyze It!. This example shows three records:

1st Record:

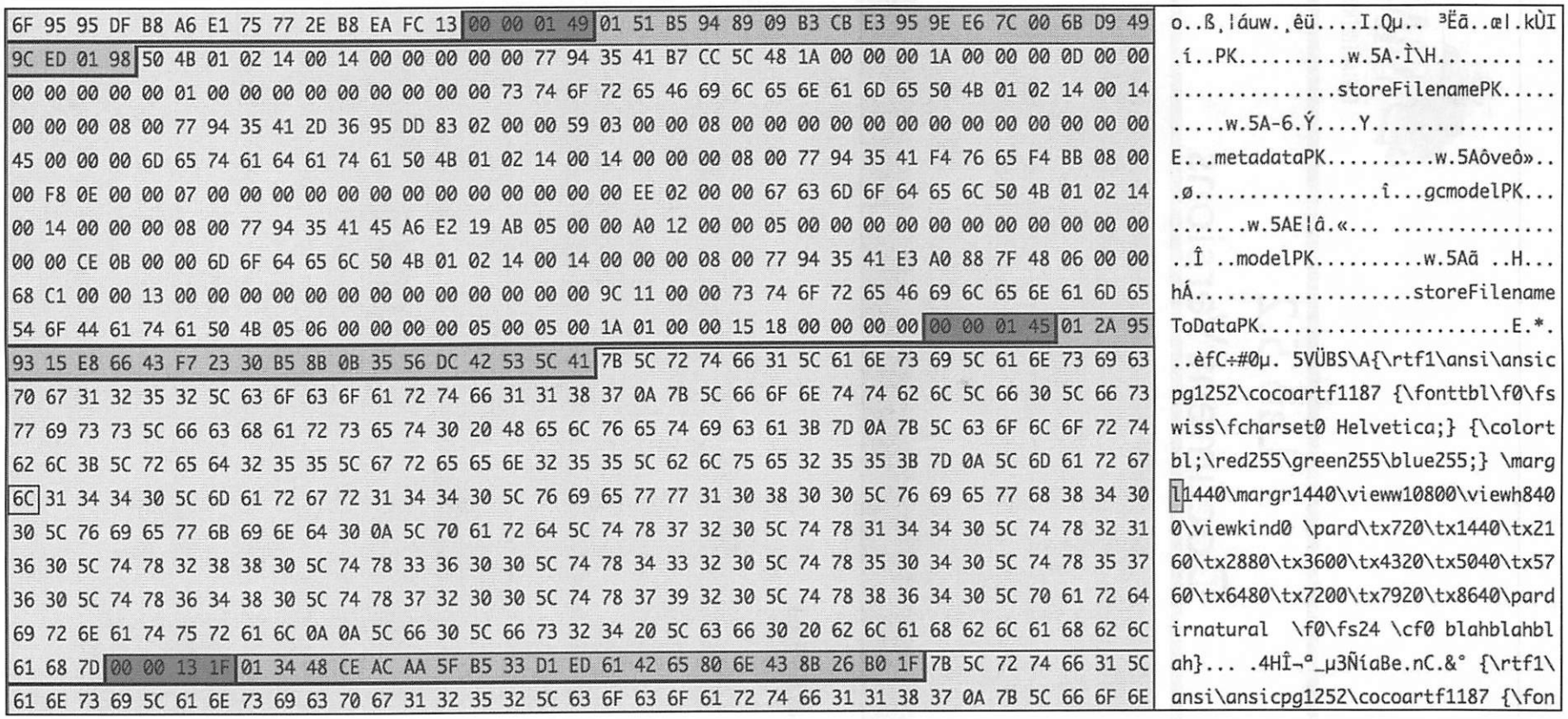
- First four bytes: Size of the chunk record—`0x00000149` (329)
- Next 21 bytes: Chunk ID (CID)—`0x0151B5948909B3CBE3959EE67C006BD9499CED0198`
- Rest of bytes: Chunk Contents

2nd Record:

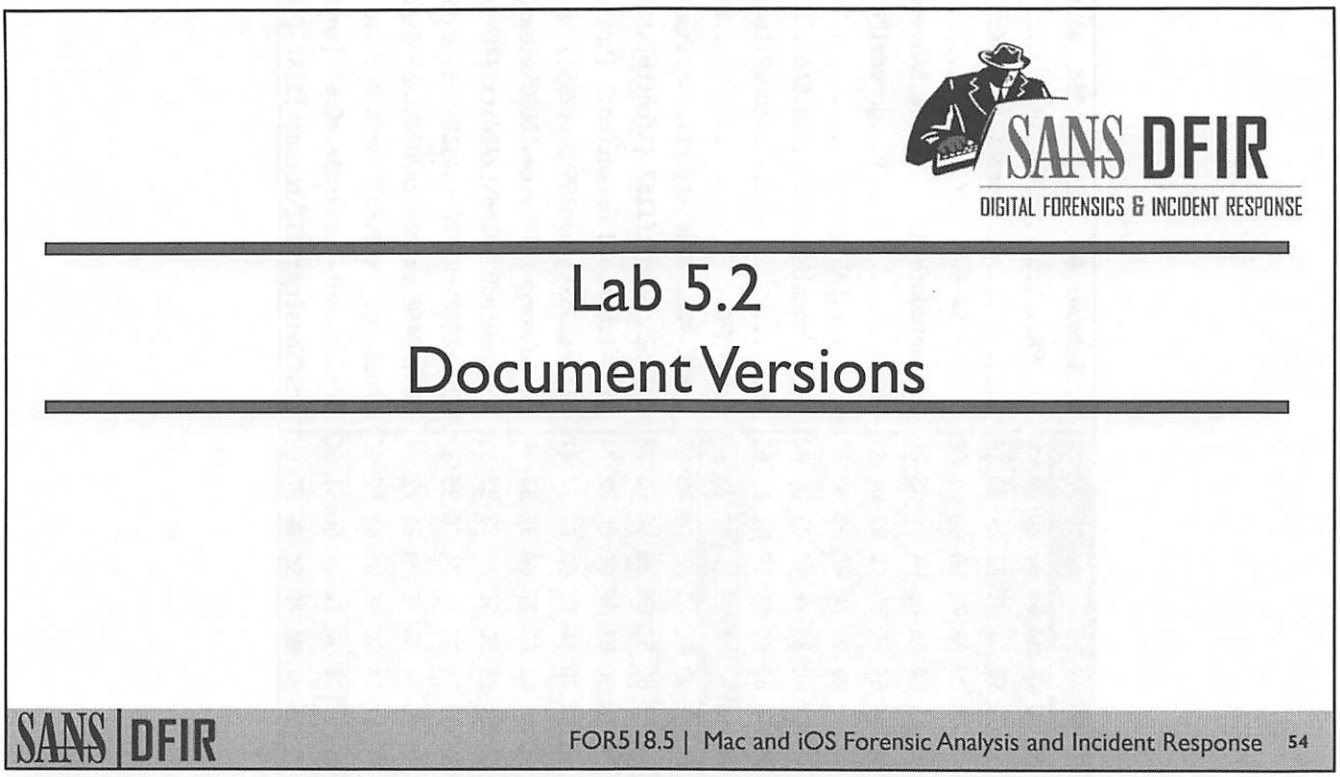
- First four bytes: Size of the chunk record—`0x00000145` (325)
- Next 21 bytes: Chunk ID (CID)—`0x012A959315E86643F72330B58B0B3556DC42535C41`
- Rest of bytes: Chunk Contents

3rd Record:

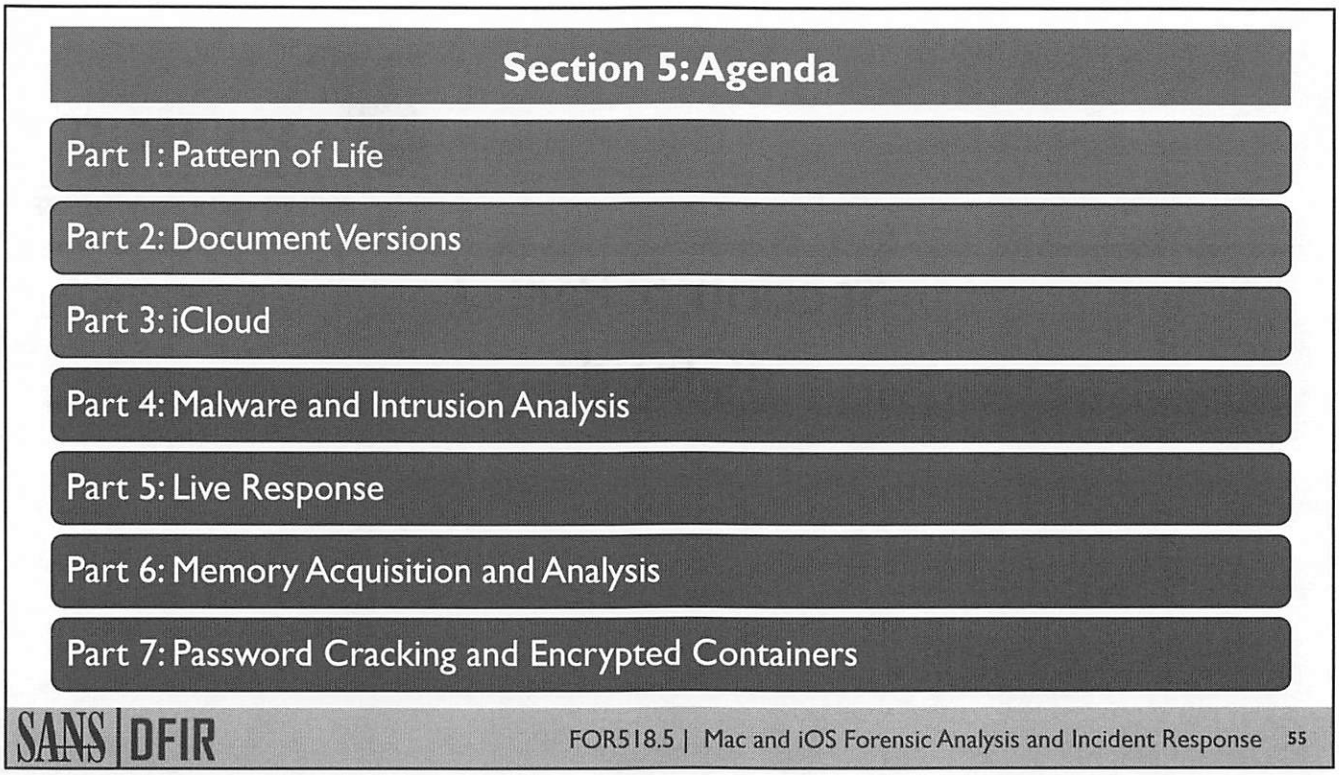
- First four bytes: Size of the chunk record—`0x0000131F` (4895)
- Next 21 bytes: Chunk ID (CID)—`0x013448CEACAA5FB533D1ED614265806E438B26B01F`
- Rest of bytes: Chunk Contents



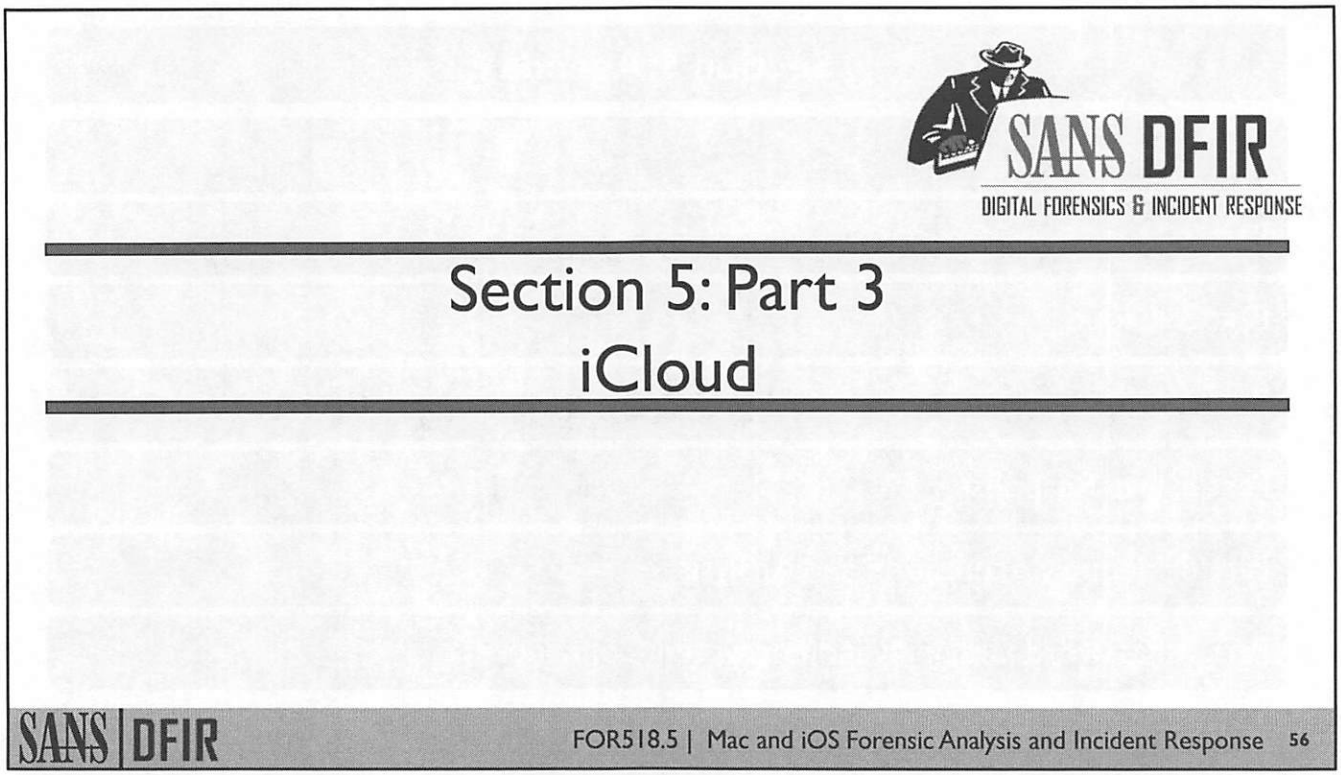
```
6F 95 95 DF B8 A6 E1 75 77 2E B8 EA FC 13 00 00 01 49 01 51 B5 94 89 09 B3 CB E3 95 9E E6 7C 00 6B D9 49  o..ß¸¦áuw.¸êü....I.Q..  ³Ëã..æ|.kÙI
9C ED 01 98 50 4B 01 02 14 00 14 00 00 00 00 00 77 94 35 41 B7 CC 5C 48 1A 00 00 00 1A 00 00 00 0D 00 00  .í..PK.........w.5A·Ì\H........ ..
00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 73 74 6F 72 65 46 69 6C 65 6E 61 6D 65 50 4B 01 02 14 00 14  ...............storeFilenamePK.....
00 00 00 08 00 77 94 35 41 2D 36 95 DD 83 02 00 00 59 03 00 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00  .....w.5A-6.Ý....Y.............
45 00 00 00 6D 65 74 61 64 61 74 61 50 4B 01 02 14 00 14 00 00 00 08 00 77 94 35 41 F4 76 65 F4 BB 08 00  E...metadataPK.........w.5Aôveô»..
00 F8 0E 00 00 07 00 00 00 00 00 00 00 00 00 00 00 00 EE 02 00 00 67 63 6D 6F 64 65 6C 50 4B 01 02 14  .ø..............î...gcmodelPK...
00 14 00 00 00 08 00 77 94 35 41 45 A6 E2 19 AB 05 00 00 A0 12 00 00 05 00 00 00 00 00 00 00 00 00 00 00  .......w.5AE¦â.«... ............
00 00 CE 0B 00 00 6D 6F 64 65 6C 50 4B 01 02 14 00 14 00 00 00 08 00 77 94 35 41 E3 A0 88 7F 48 06 00 00  ..Î ..modelPK.........w.5Aã ..H...
68 C1 00 00 13 00 00 00 00 00 00 00 00 00 00 00 00 00 9C 11 00 00 73 74 6F 72 65 46 69 6C 65 6E 61 6D 65  hÁ..................storeFilename
54 6F 44 61 74 61 50 4B 05 06 00 00 00 00 05 00 05 00 1A 01 00 00 15 18 00 00 00 00 00 00 01 45 01 2A 95  ToDataPK.....................E.*.
93 15 E8 66 43 F7 23 30 B5 8B 0B 35 56 DC 42 53 5C 41 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73 69 63  ..èfC÷#0µ. 5VÜBS\A{\rtf1\ansi\ansic
70 67 31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38 37 0A 7B 5C 66 6F 6E 74 74 62 6C 5C 66 30 5C 66 73  pg1252\cocoartf1187 {\fonttbl\f0\fs
77 69 73 73 5C 66 63 68 61 72 73 65 74 30 20 48 65 6C 76 65 74 69 63 61 3B 7D 0A 7B 5C 63 6F 6C 6F 72 74  wiss\fcharset0 Helvetica;} {\colort
62 6C 3B 5C 72 65 64 32 35 35 5C 67 72 65 65 6E 32 35 35 5C 62 6C 75 65 32 35 35 3B 7D 0A 5C 6D 61 72 67  bl;\red255\green255\blue255;} \marg
6C 31 34 34 30 5C 6D 61 72 67 72 31 34 34 30 5C 76 69 65 77 77 31 30 38 30 30 5C 76 69 65 77 68 38 34 30  l1440\margr1440\vieww10800\viewh840
30 5C 76 69 65 77 6B 69 6E 64 30 0A 5C 70 61 72 64 5C 74 78 37 32 30 5C 74 78 31 34 34 30 5C 74 78 32 31  0\viewkind0 \pard\tx720\tx1440\tx21
36 30 5C 74 78 32 38 38 30 5C 74 78 33 36 30 30 5C 74 78 34 33 32 30 5C 74 78 35 30 34 30 5C 74 78 35 37  60\tx2880\tx3600\tx4320\tx5040\tx57
36 30 5C 74 78 36 34 38 30 5C 74 78 37 32 30 30 5C 74 78 37 39 32 30 5C 74 78 38 36 34 30 5C 70 61 72 64  60\tx6480\tx7200\tx7920\tx8640\pard
69 72 6E 61 74 75 72 61 6C 0A 0A 5C 66 30 5C 66 73 32 34 20 5C 63 66 30 20 62 6C 61 68 62 6C 61 68 62 6C  irnatural  \f0\fs24 \cf0 blahblahbl
61 68 7D 00 00 13 1F 01 34 48 CE AC AA 5F B5 33 D1 ED 61 42 65 80 6E 43 8B 26 B0 1F 7B 5C 72 74 66 31 5C  ah}... .4HÎ¬ª_µ3ÑíaBe.nC.&° {\rtf1\
61 6E 73 69 5C 61 6E 73 69 63 70 67 31 32 35 32 5C 63 6F 63 6F 61 72 74 66 31 31 38 37 0A 7B 5C 66 6F 6E  ansi\ansicpg1252\cocoartf1187 {\fon
```



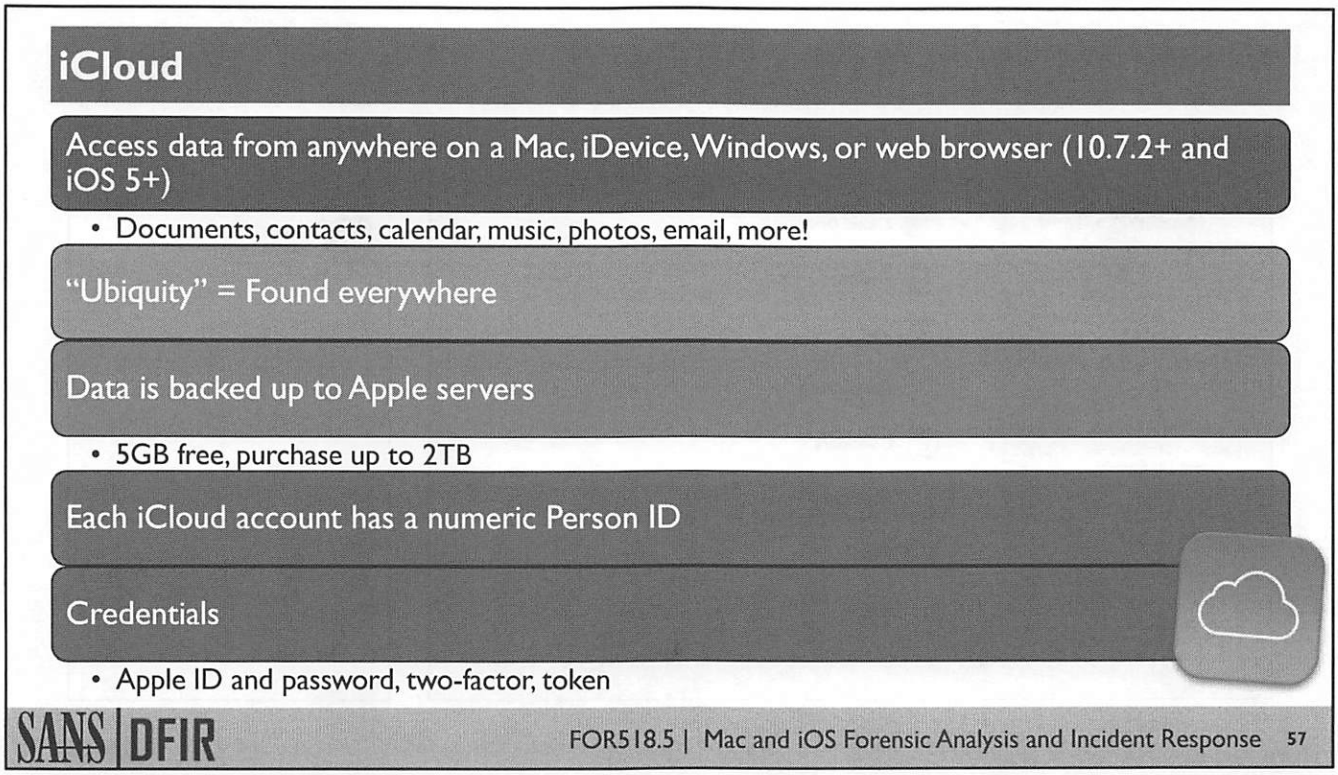
This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.

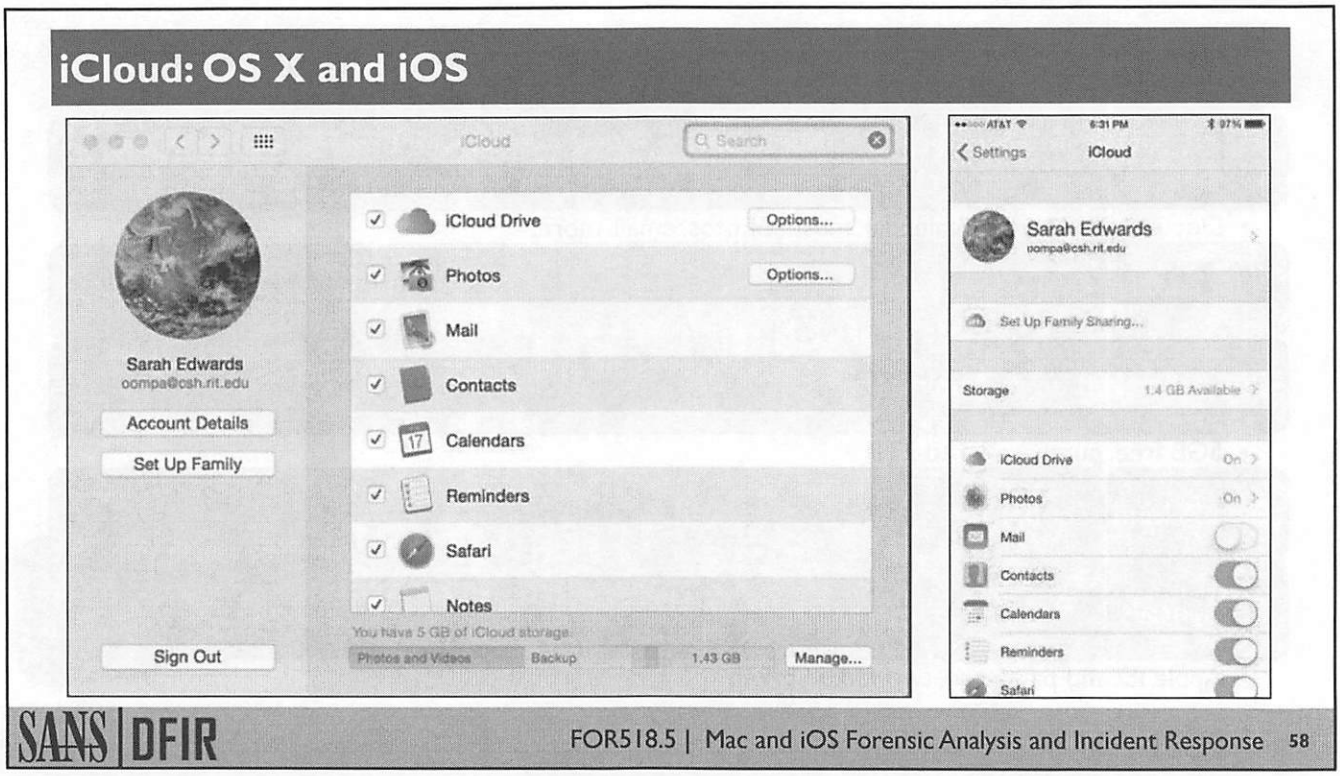


iCloud is the term used for Apple's cloud services. iCloud has the ability to access data anywhere, including documents, contacts, calendar, music, photos, and email.

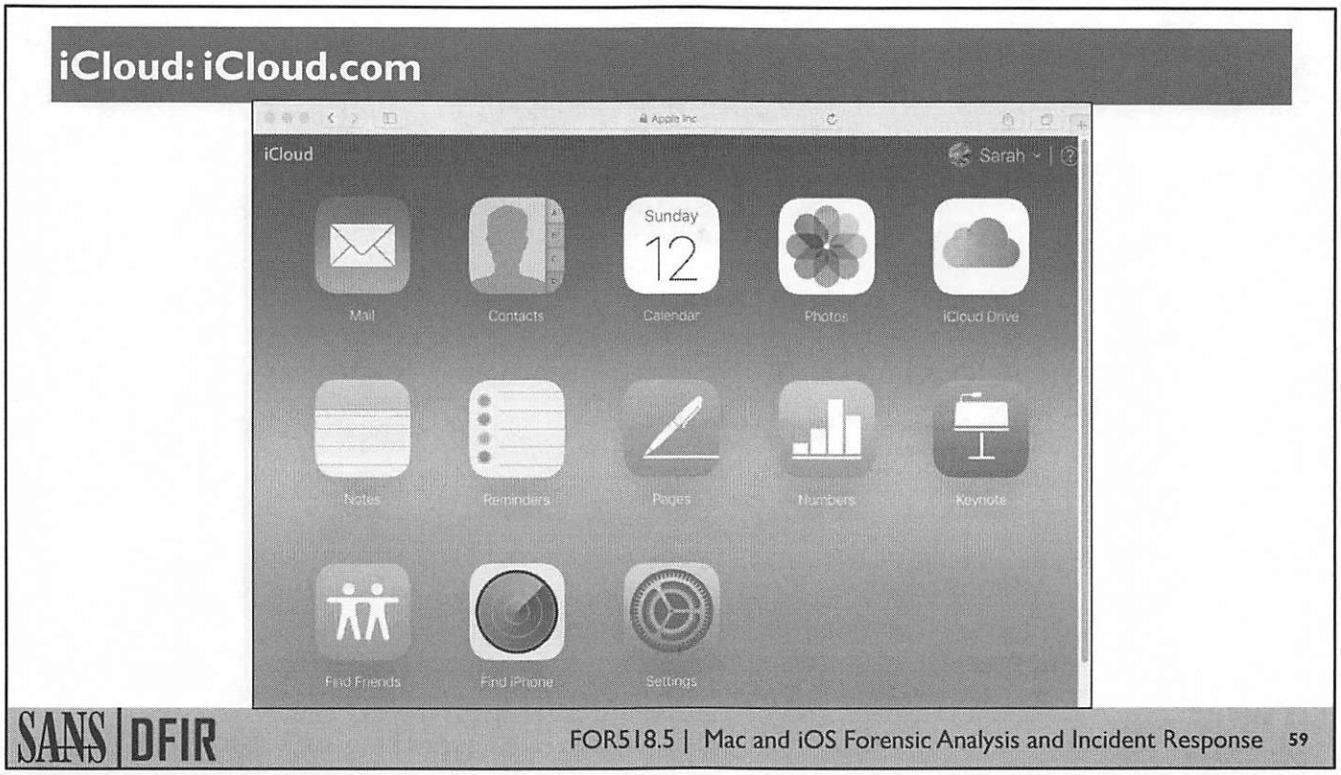
iCloud can be accessed natively from any OS X system (10.7.2+), iOS device (5+), or from a web browser (via iCloud.com). This data is stored on Apple servers. Rumor is that they may be using Amazon and Microsoft cloud servers on the backend.

Each iCloud account uses a unique identifier known as a Person ID.

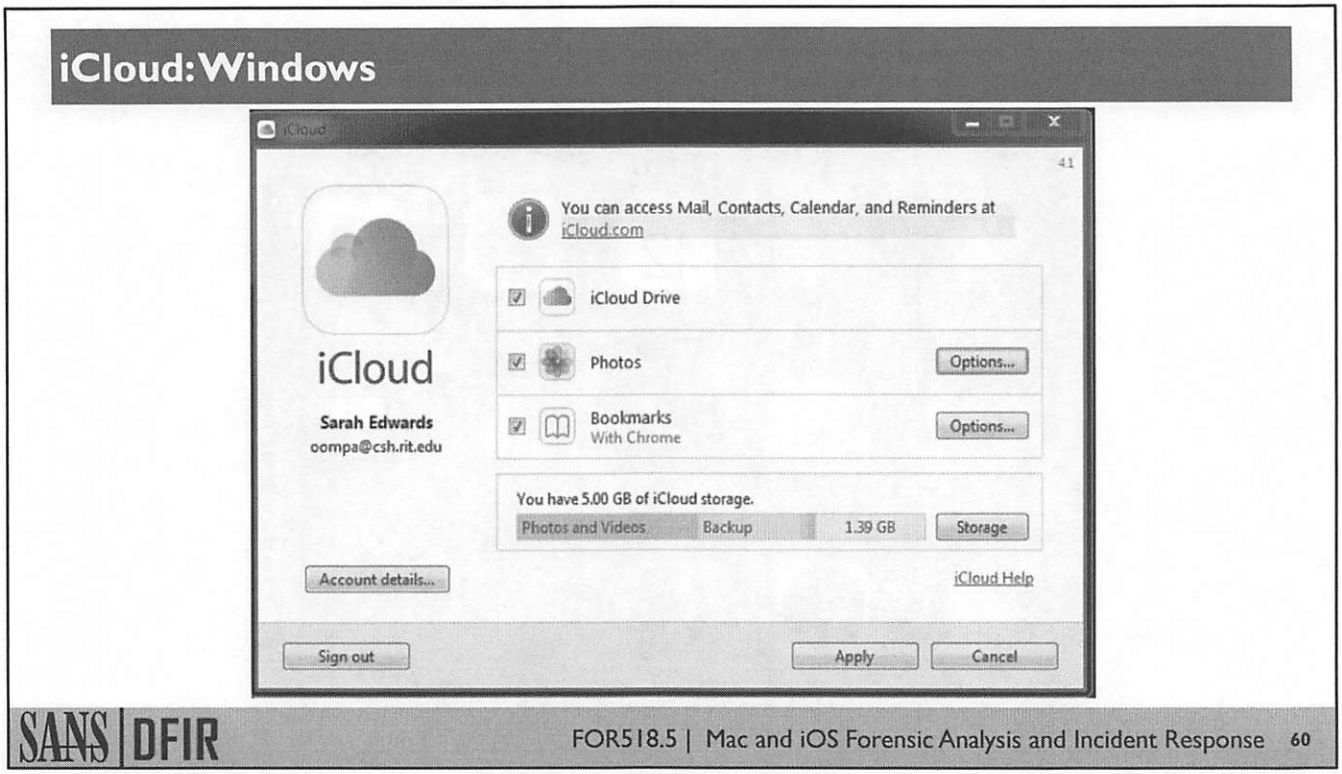
One term you will see often in almost everything related to iCloud is "Ubiquity", which means "found everywhere". iCloud documents, email, contacts, etc. are meant to be accessible everywhere.



This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.

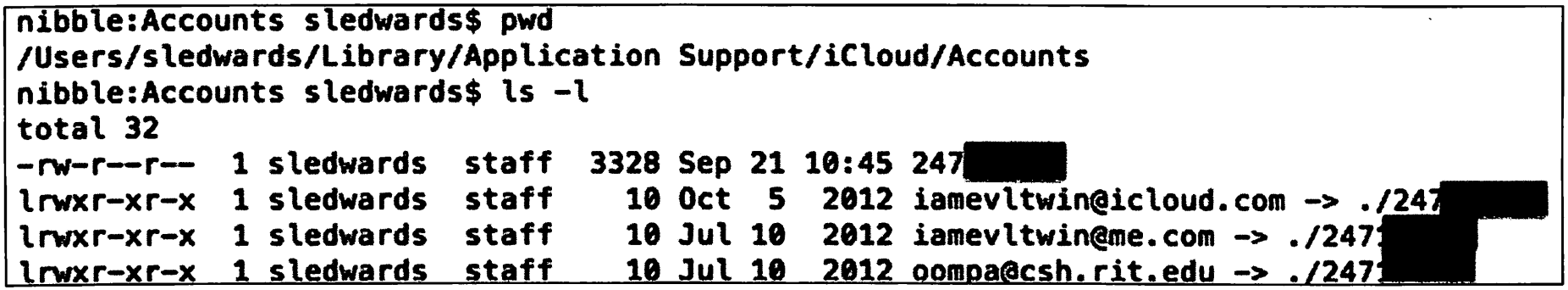
iCloud: Synced Accounts ~/Library/Application Support/iCloud/Accounts/

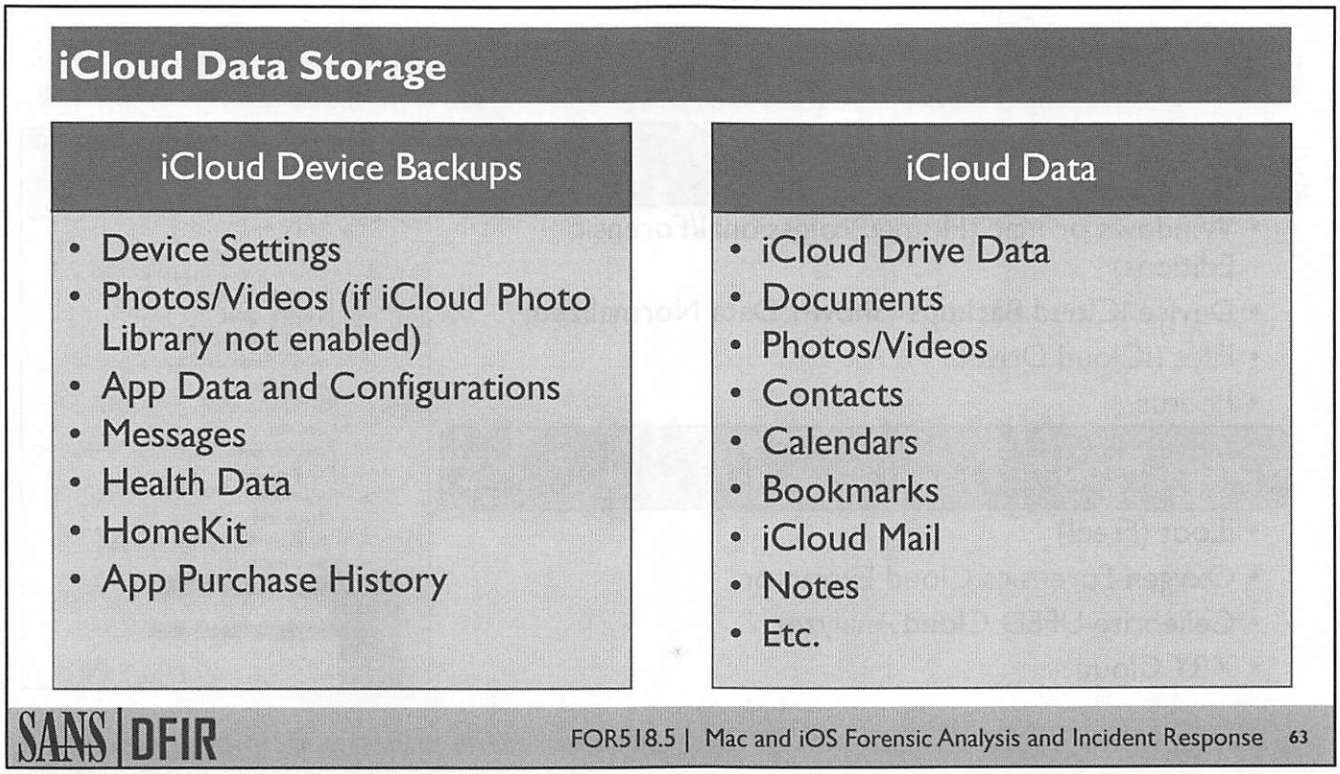
```
nibble:Accounts sledwards$ pwd
/Users/sledwards/Library/Application Support/iCloud/Accounts
nibble:Accounts sledwards$ ls -l
total 32
-rw-r--r--  1 sledwards  staff  3328 Sep 21 10:45 247[redacted]
lrwxr-xr-x  1 sledwards  staff    10 Oct  5  2012 iamevltwin@icloud.com -> ./247[redacted]
lrwxr-xr-x  1 sledwards  staff    10 Jul 10  2012 iamevltwin@me.com -> ./247[redacted]
lrwxr-xr-x  1 sledwards  staff    10 Jul 10  2012 oompa@csh.rit.edu -> ./247[redacted]
```


The iCloud Person ID number is the unique number associated with a user's iCloud account.

Shown in the screenshot above are the files located in the user's `~/Library/Application Support/iCloud/Accounts/` directory. This directory contains a file with a numeric filename. This number is the user's iCloud Person ID. This iCloud ID has been redacted for privacy reasons.

This directory also contains link files that point to the numeric filename. Each link is an email associated with that user's iCloud account. This iCloud account has been associated with three email accounts.





iCloud stores its data in a few different "places". When using different tools to pull down this data, you may or may not be getting all the iCloud data.

Reference:
`https://support.apple.com/en-us/HT207428`

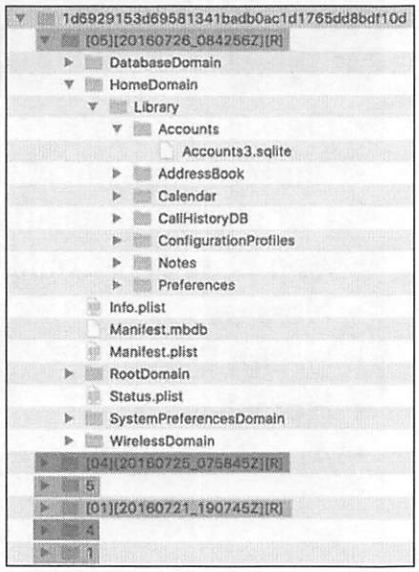
iCloud Data Extraction

Elcomsoft Phone Breaker

- Windows or Mac (Home/Professional/Forensic Editions)
- Device iCloud Backups (Shown, Data Normalized)
- Files (iCloud Drive)
- Photos

Other Tools:

- iLoot (Free!)
- Oxygen Forensics Cloud Extractor
- Cellebrite UFED Cloud Analyzer
- XRY Cloud

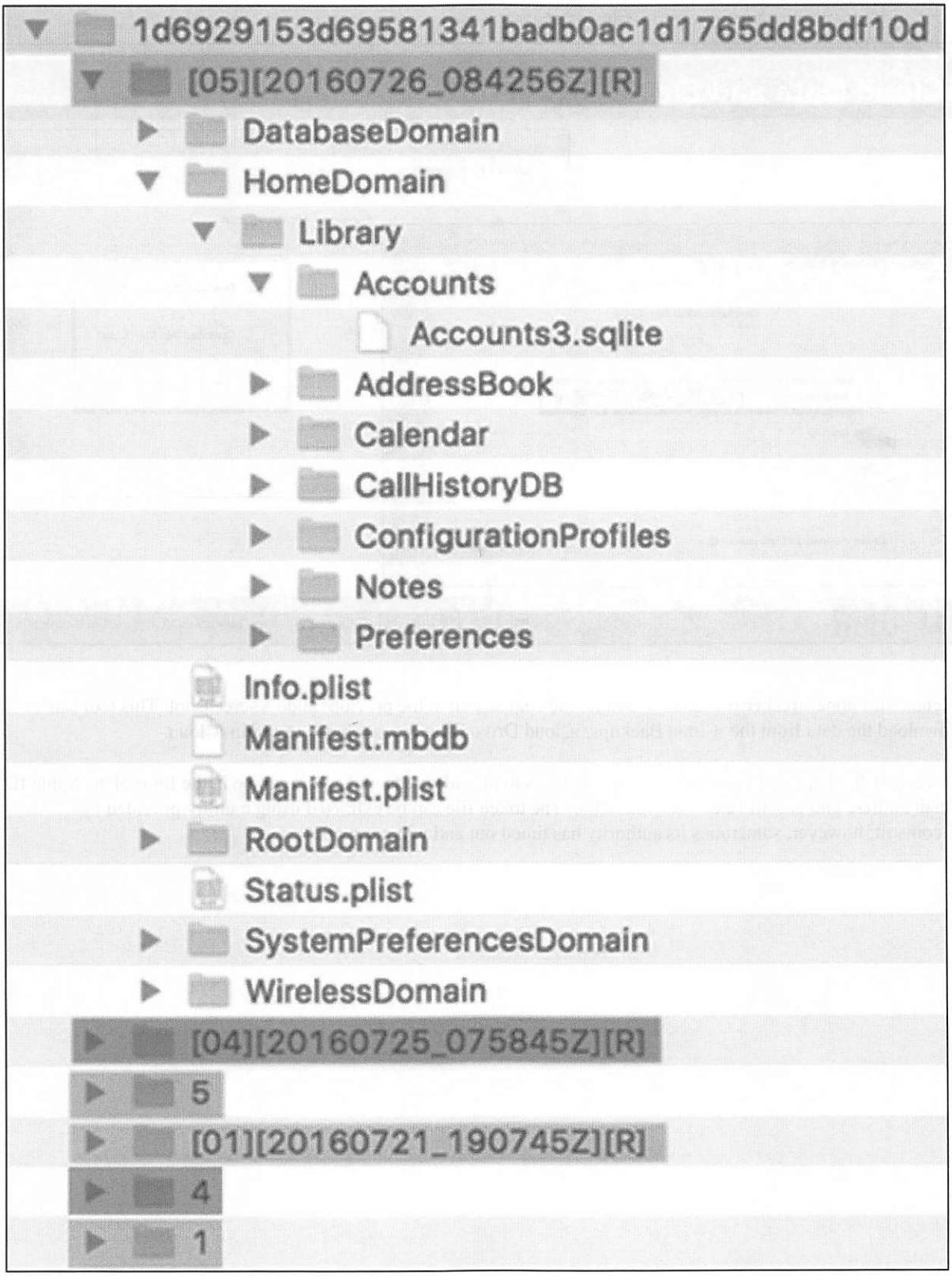


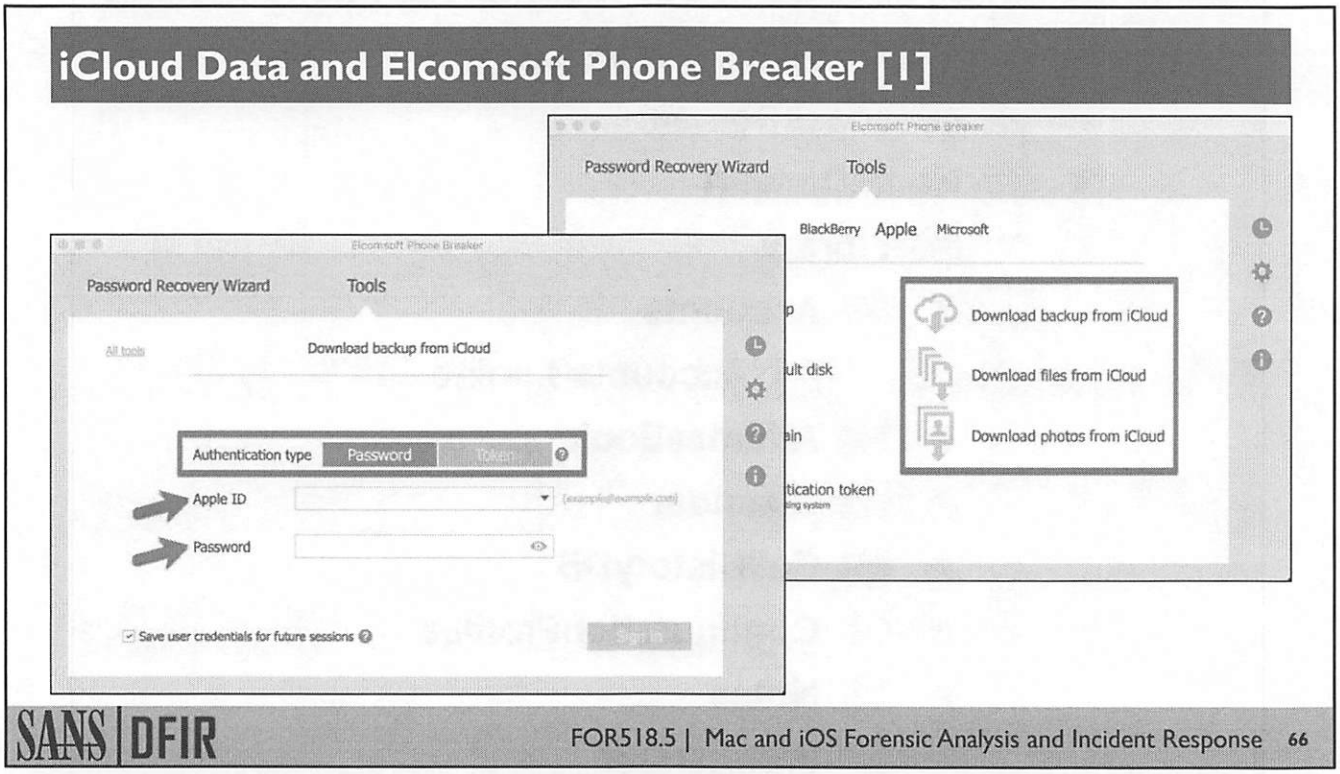
iCloud files might be available for download from Apple's servers assuming you have the credentials and authority required. Apple has been shoring up its iCloud data security since some high-profile celebrity hacks. Access to the iCloud data will almost definitely change in the future. Apple has been changing the iCloud communications protocol, which means the tools also need to update.

As of this writing, Elcomsoft Phone Breaker was able to pull down iCloud data. This data can include iCloud iDevice backups, as well as iCloud Drive data and photos.

The screenshot on the right shows an example of what Elcomsoft Phone Breaker will extract. The directories labeled '1', '4', and '5' are non-normalized data; the directories labeled '[0#][DATE_TIME][R]' are normalized. The normalized data as shown is analyst friendly—you can view files as they might have been stored in an iTunes-style backup.

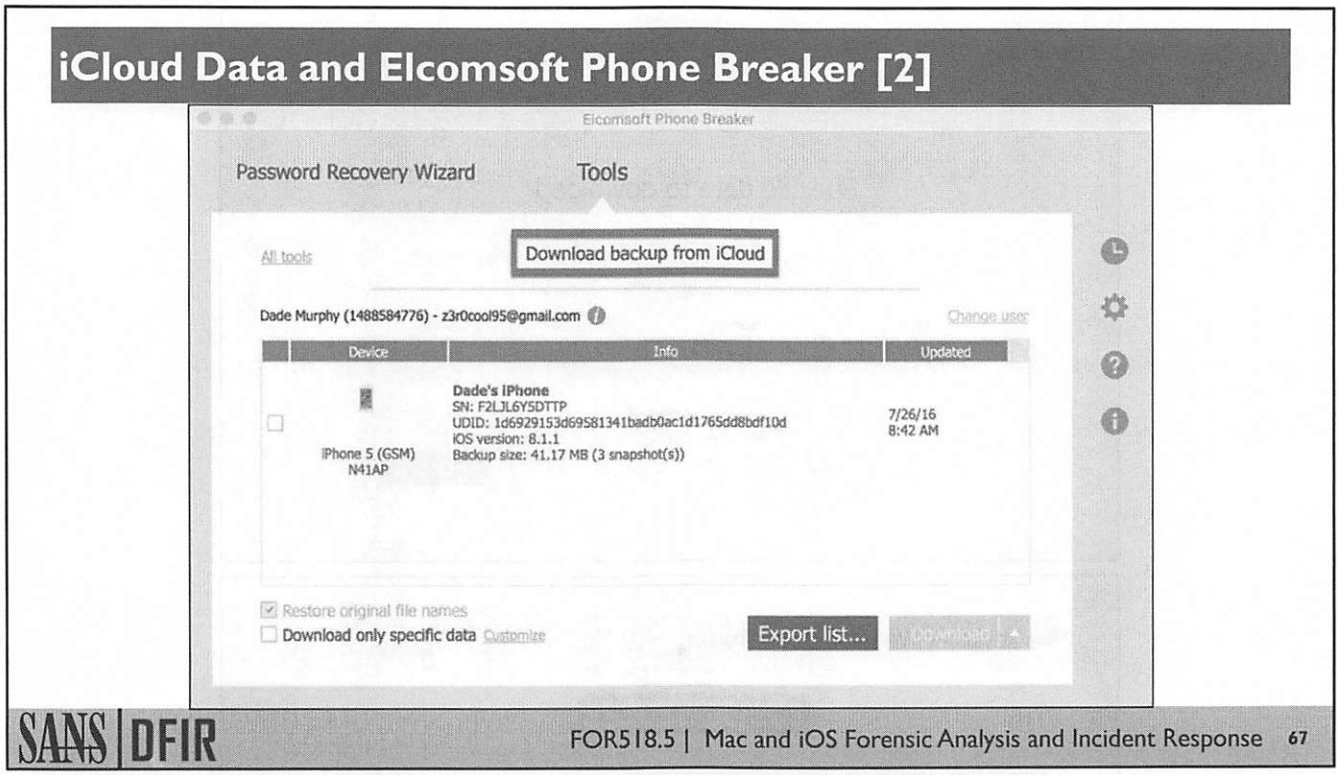
References:
`https://www.elcomsoft.com/eppb.html`
`http://www.oxygen-forensic.com/en/products/oxygen-forensic-detective/detective/cloud-data-extraction`
`http://www.cellebrite.com/Mobile-Forensics/Products/ufed-cloud-analyzer`
`https://www.msab.com/products/xry/#_cloud`





Elcomsoft Phone Breaker was used to extract the data shown in the previous slide's screenshot. This tool can download the data from the iCloud Backups, iCloud Drive files, or just photos stored in iCloud.

Once an iCloud option is selected, the analyst needs to provide credentials. This can be in the form of an Apple ID email address and password or via a token file. The token file can be extracted using a utility provided by Elcomsoft; however, sometimes its authority has timed out and will not work.

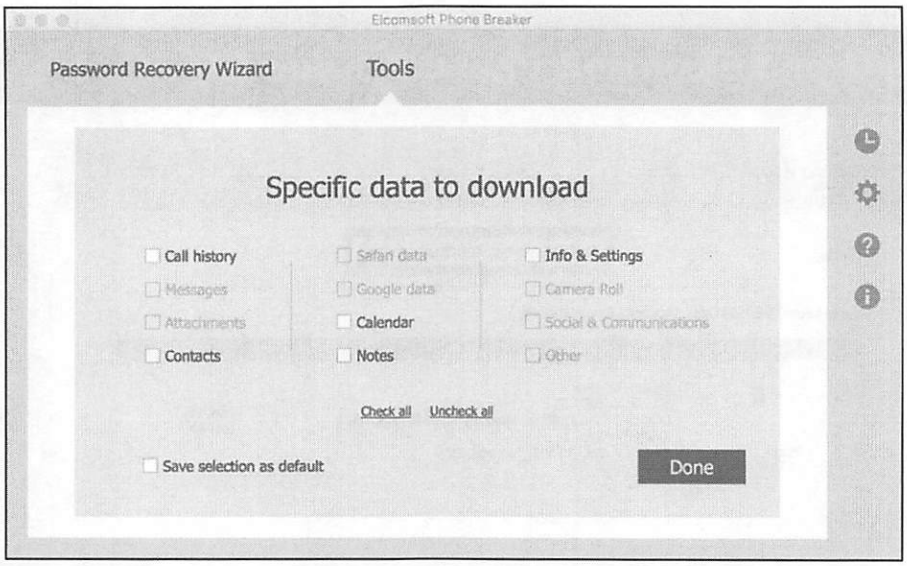


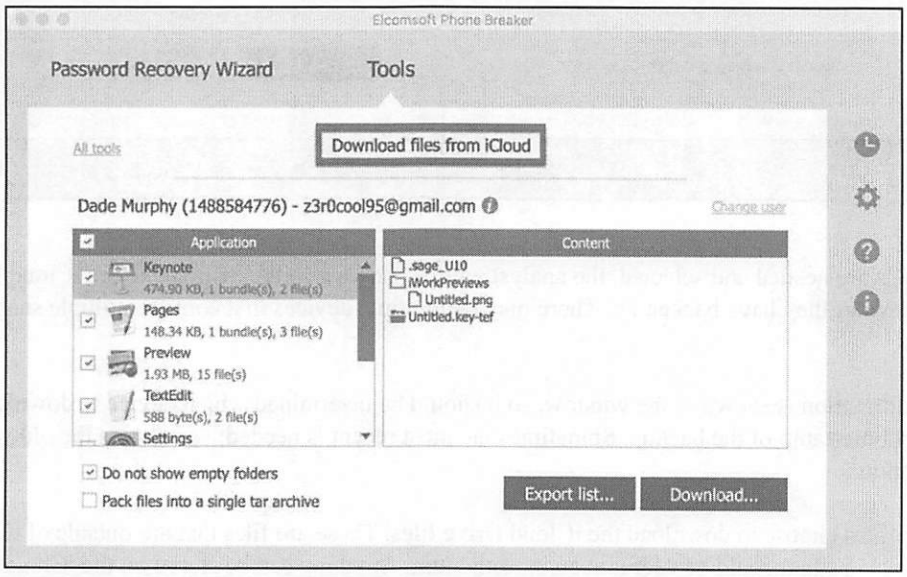
If iCloud Backups are needed and selected, the analyst will be shown a screen with the user's iCloud Backup files for whichever devices they have backed up. There may be multiple devices that contain multiple snapshots or backups of data.

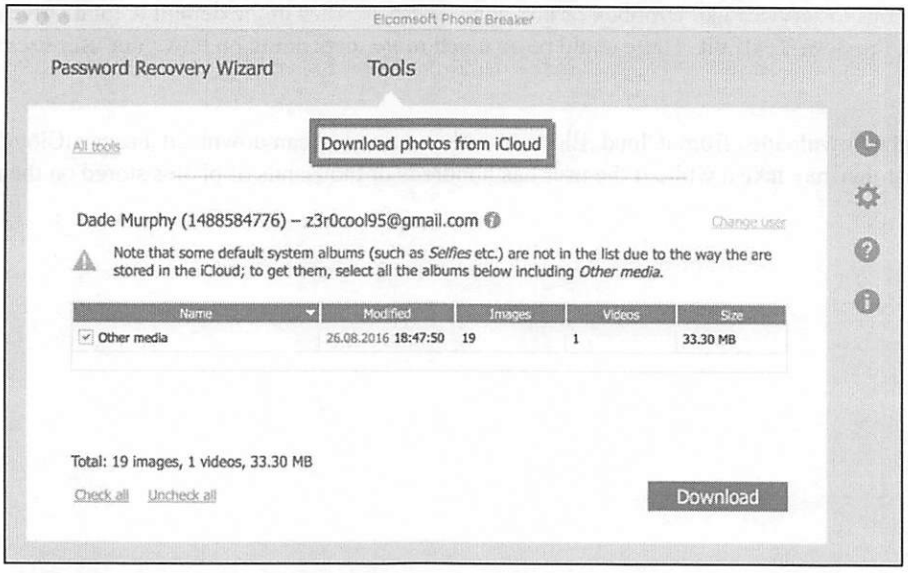
The device's information is shown in the window, so it should be determined which device to download by legal authority and by timestamp of the backup. Sometimes the most recent is needed; sometimes the oldest—depending on the investigation.

The analyst may also choose to download the iCloud Drive files. These are files that are outside of the normal iCloud device backups that could be useful in an investigation. Anything can be stored on the iCloud Drive. The service is analogous to services like Dropbox or box.com. There are files in the default iCloud Drive directories: Keynote, Pages, Preview, TextEdit. There could be so much more, depending on how your user uses their iCloud Drive.

Photos can also be downloaded from iCloud. Elcomsoft Phone Breaker can download a user's iCloud photos. It is worth noting that this may take a while if the user has hundreds or thousands of photos stored on their iCloud accounts.







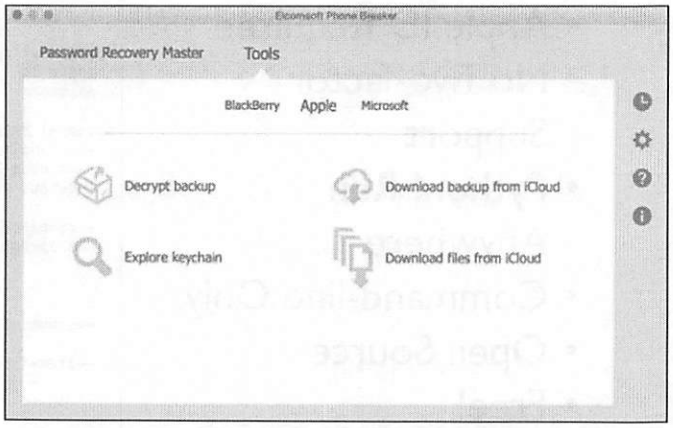
Getting to the iCloud Data: iLoot

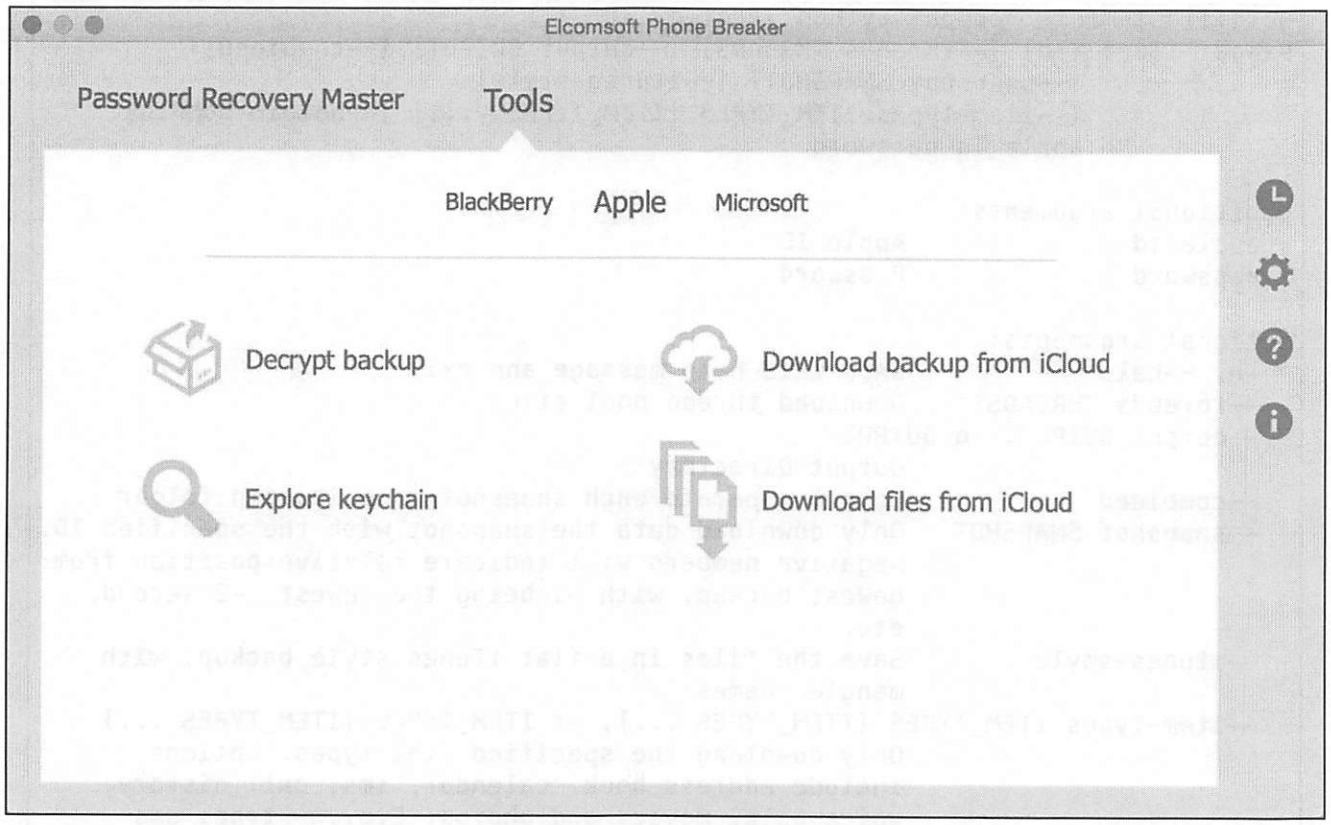
- iLoot:
  - Apple ID Required
  - No Two-factor Support
  - Python! Run Anywhere
  - Command-line Only
  - Open Source
  - Free!

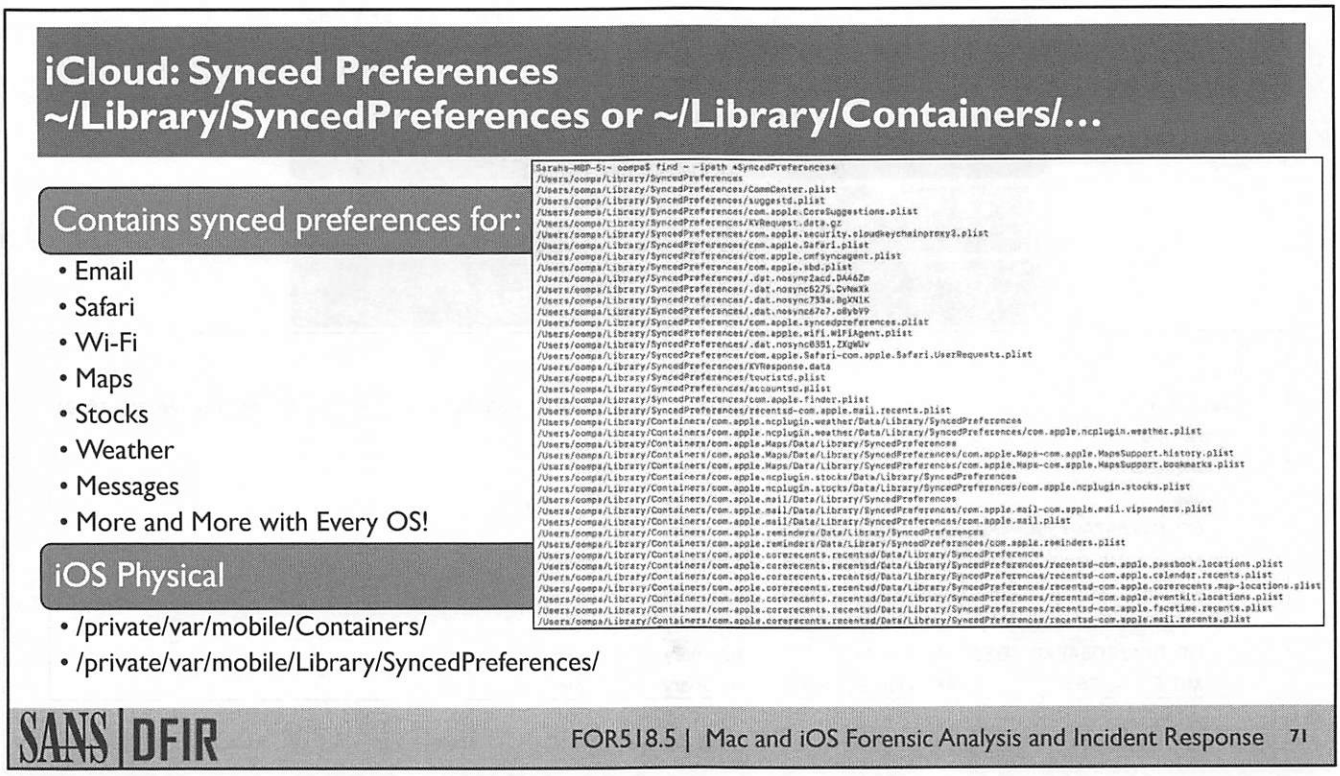
```
word:iloot-master oompa$ python iloot.py -h
usage: iloot [-h] [--threads THREADS] [--output OUTPUT] [--combined]
             [--snapshot SNAPSHOT] [--itunes-style]
             [--item-types ITEM_TYPES [ITEM_TYPES ...]] [--domain DOMAIN]
             apple_id password

positional arguments:
  apple_id              Apple ID
  password              Password

optional arguments:
  -h, --help            show this help message and exit
  --threads THREADS     Download thread pool size
  --output OUTPUT, -o OUTPUT
                        Output Directory
  --combined            Do not separate each snapshot into its own folder
  --snapshot SNAPSHOT   Only download data the snapshot with the specified ID.
                        Negative numbers will indicate relative position from
                        newest backup, with -1 being the newest, -2 second,
                        etc.
  --itunes-style        Save the files in a flat iTunes-style backup, with
                        mangled names
  --item-types ITEM_TYPES [ITEM_TYPES ...], -t ITEM_TYPES [ITEM_TYPES ...]
                        Only download the specified item types. Options
                        include address_book, calendar, sms, call_history,
                        voicemails, movies and photos. E.g., --types sms
                        voicemail
  --domain DOMAIN, -d DOMAIN
                        Limit files to those within a specific application
                        domain
```

Getting to the iCloud Data: Elcomsoft EPB

- Elcomsoft Phone Breaker (EPB)
  - "Forensic"
  - Apple ID or Authentication Token
  - Support for Two-factor
  - Mac or Windows
  - Professional or Forensic Editions
  - iCloud Backups and iCloud Files (iCloud Drive)
  - $200, $800







iCloud sync preferences are stored in the `SyncedPreferences` directories. These are the technical details on how iCloud syncs with the Apple servers to sync the data across all devices.

The main configuration data can be found in the `com.apple.syncedpreferences.plist` file.

Certain applications each have their own property list with their associated preferences. These files may be located in either the `~/Library/SyncedPreferences/` directory itself or in the sandbox container directory located in the `~/Library/Containers/<bundle_id>/Data/Library/SyncedPreferences/`.

```
Sarahs-MBP-5:~ oompa$ find ~ -ipath *SyncedPreferences*
/Users/oompa/Library/SyncedPreferences
/Users/oompa/Library/SyncedPreferences/CommCenter.plist
/Users/oompa/Library/SyncedPreferences/suggestd.plist
/Users/oompa/Library/SyncedPreferences/com.apple.CoreSuggestions.plist
/Users/oompa/Library/SyncedPreferences/KVRequest.data.gz
/Users/oompa/Library/SyncedPreferences/com.apple.security.cloudkeychainproxy3.plist
/Users/oompa/Library/SyncedPreferences/com.apple.Safari.plist
/Users/oompa/Library/SyncedPreferences/com.apple.cmfsyncagent.plist
/Users/oompa/Library/SyncedPreferences/com.apple.sbd.plist
/Users/oompa/Library/SyncedPreferences/.dat.nosync2acd.DA46Zm
/Users/oompa/Library/SyncedPreferences/.dat.nosync5275.CvNwXk
/Users/oompa/Library/SyncedPreferences/.dat.nosync733a.BgXNlK
/Users/oompa/Library/SyncedPreferences/.dat.nosync67c7.o8ybV9
/Users/oompa/Library/SyncedPreferences/com.apple.syncedpreferences.plist
/Users/oompa/Library/SyncedPreferences/com.apple.wifi.WiFiAgent.plist
/Users/oompa/Library/SyncedPreferences/.dat.nosync0351.ZXgWUv
/Users/oompa/Library/SyncedPreferences/com.apple.Safari-com.apple.Safari.UserRequests.plist
/Users/oompa/Library/SyncedPreferences/KVResponse.data
/Users/oompa/Library/SyncedPreferences/touristd.plist
/Users/oompa/Library/SyncedPreferences/accountsd.plist
/Users/oompa/Library/SyncedPreferences/com.apple.finder.plist
/Users/oompa/Library/SyncedPreferences/recentsd-com.apple.mail.recents.plist
/Users/oompa/Library/Containers/com.apple.ncplugin.weather/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.ncplugin.weather/Data/Library/SyncedPreferences/com.apple.ncplugin.weather.plist
/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.history.plist
/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.bookmarks.plist
/Users/oompa/Library/Containers/com.apple.ncplugin.stocks/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.ncplugin.stocks/Data/Library/SyncedPreferences/com.apple.ncplugin.stocks.plist
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/SyncedPreferences/com.apple.mail-com.apple.mail.vipsenders.plist
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/SyncedPreferences/com.apple.mail.plist
/Users/oompa/Library/Containers/com.apple.reminders/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.reminders/Data/Library/SyncedPreferences/com.apple.reminders.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.passbook.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.calendar.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.corerecents.map-locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.eventkit.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.facetime.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.mail.recents.plist
```

Synced Preferences: Email—Recent Emails [1]

*.-com.apple.mail.recents.plist

MR_* – Person-to-Person Email

GP_* – Group Email

Key	Type	Value
▼ Root	Dictionary	(4 items)
versionid	String	FT=-@RU=d8326b9f-4f49-46cb-8e6f-99bb3e963f9b@S=78594
initialsync	Number	2
changecount	Number	491
▼ values	Dictionary	(680 items)
▶ GP_D42A57A67E5DF4FDB4BDC67B9FF2D9C7	Dictionary	(3 items)
▶ MR_5DA23DA86B5574B5C862872558D6E2C8	Dictionary	(3 items)
▶ MR_1ABCF8ACFDD90B8EA7A46833DEFAA2F4	Dictionary	(3 items)
▶ MR_9CFDBEF0E4FC5C54D4731E49AF12C323	Dictionary	(3 items)
▶ GP_B643BD5AEA804BE04C1ADF43A7F30311	Dictionary	(3 items)
▶ MR_C9DAFF659EDC27A0FA6E8094787A33A7	Dictionary	(3 items)

macOS:

- `~/Library/SyncedPreferences/com.apple.mail-com.apple.mail.recents.plist`
- `~/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.mail.recents.plist`

iOS:

`/private/var/mobile/Library/SyncedPreferences/com.apple.cloudrecents.CloudRecentsAgent-com.apple.mail.recents.plist`

The synced preferences for the Mail application are found in a few property lists. The one shown above is the `com.apple.mail-com.apple.mail.recents.plist` property list file.

This property list contains data about the "recent" email contacts. Fortunately for us, in reality it contains even not-so-recent email contacts! Note the value count for the `values` key!

Each key under the values key (`MR_*/GP_*`) contains information pertaining to an email contact with multiple keys.

- `t`: Timestamps of previous emails
- `n`: Contact Name
- `a`: Contact's Email Address

Synced Preferences: Email—Recent Emails [2]

MR_9CFDBEF0E4FC5C54D4731E49AF12C323	Dictionary	(3 items)
value	Dictionary	(6 items)
S	String	com.apple.mail
v	Number	1
n	String	Heather Mahalik
s	String	oompa@csh.rit.edu
a	String	hmahalik@gmail.com
t	Array	(5 items)
Item 0	Date	May 12, 2015, 9:30:38 PM
Item 1	Date	May 10, 2015, 2:40:35 PM
Item 2	Date	May 10, 2015, 2:23:48 PM
Item 3	Date	May 10, 2015, 12:21:41 PM
Item 4	Date	May 6, 2015, 7:52:22 PM
remotevalue	Data	<0179770f bee0021b 00000
timestamp	Number	453,173,438

GP_B643BD5AEA804BE04C1ADF43A7F30311	Dictionary	(3 items)
value	Dictionary	(8 items)
k	String	gr
t	Array	(3 items)
Item 0	Date	Mar 12, 2015, 9:47:43 PM
Item 1	Date	Jul 28, 2014, 5:44:23 PM
Item 2	Date	Jul 27, 2014, 10:26:47 AM
gK	Number	0
S	String	com.apple.mail
v	Number	1
n	String	Rob Lee
s	String	oompa@csh.rit.edu
mrs	Array	(2 items)
Item 0	Dictionary	(3 items)
n	String	Henri van Goethem
a	String	hvangoethem@sans.org
k	String	email
Item 1	Dictionary	(3 items)
n	String	Rob Lee
a	String	rlee@sans.org
k	String	email
remotevalue	Data	<01196a02 3f79b21a 00000
timestamp	Number	447,904,063

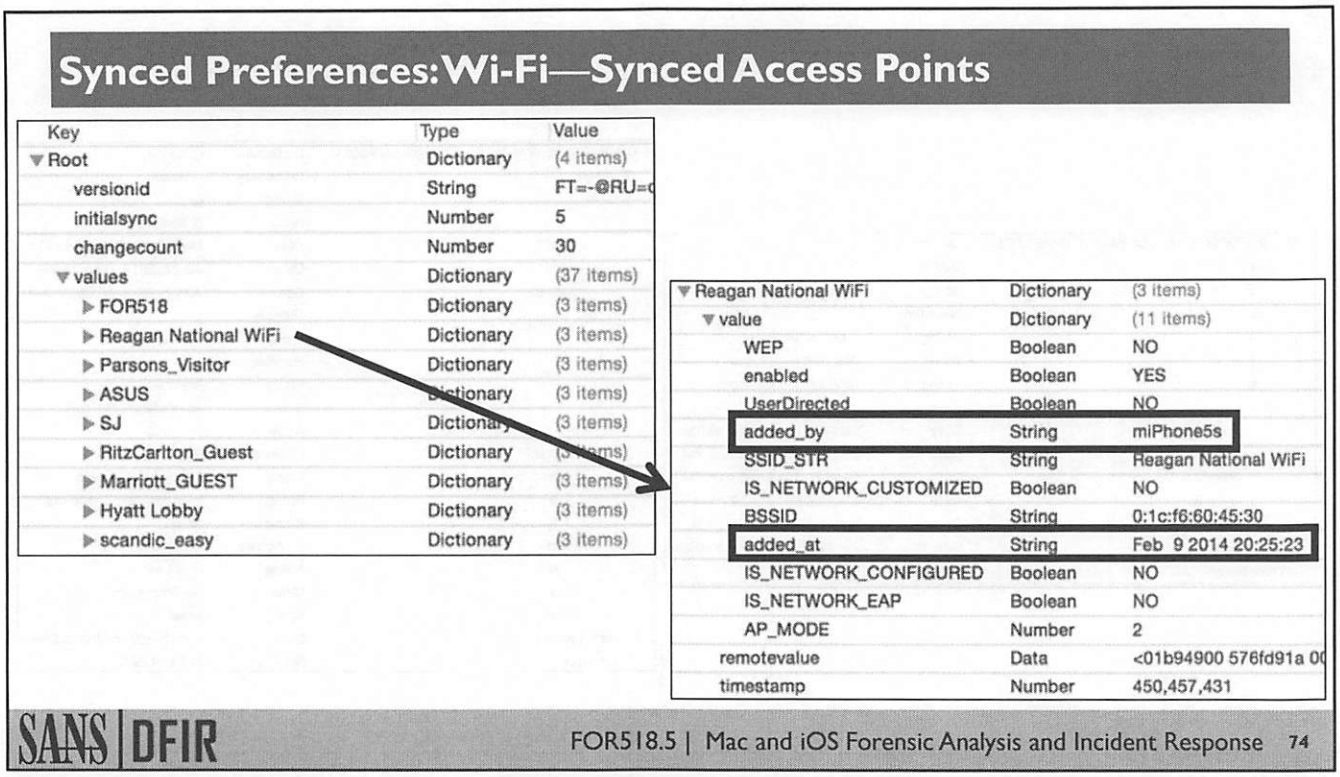
The synced preferences for the Mail application are found in a few property lists. The one shown above is the `com.apple.mail-com.apple.mail.recents.plist` property list file.

This property list contains data about the "recent" email contacts. Fortunately for us, in reality it contains even not-so-recent email contacts! Note the value count for the `values` key!

Each key under the values key (`MR_*`) contains information pertaining to an email contact with multiple keys.

- `t`: Timestamps of previous emails
- `n`: Contact Name
- `a` : Contact's Email Address

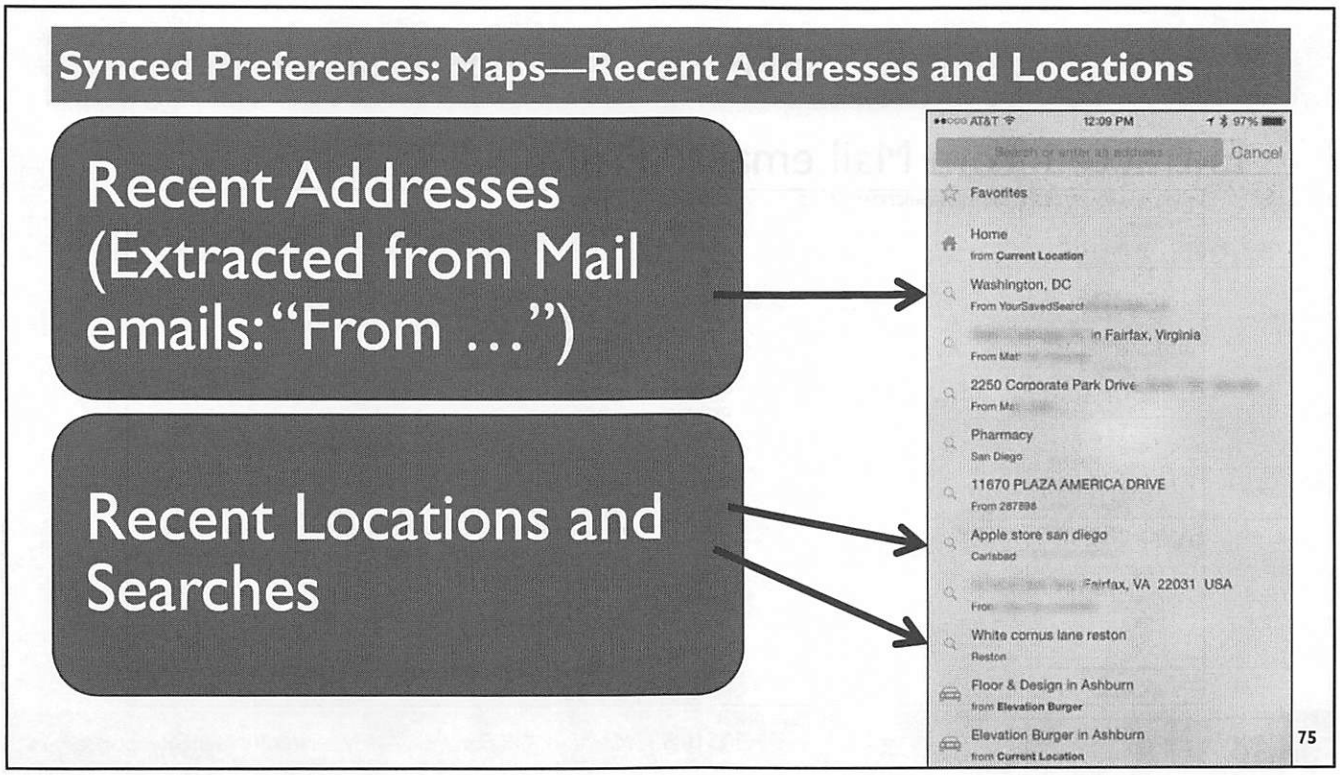
The synced preferences for the Mail application are found in a few property lists. The one shown above is the `com.apple.mail-com.apple.mail.recents.plist` property list file.



macOS: `~/Library/SyncedPreferences/com.apple.wifi.WiFiAgent.plist`
iOS: `/private/var/mobile/Library/SyncedPreferences/com.apple.wifid.plist`

Along with the native access point information in the `com.apple.airport.plist` file, iCloud will sync some access point information. It is worth noting that these have additional keys.

`added_by`: The device name that added this access point.
`added_at`: The time it was added. Warning: Not necessarily added when it was connected to.



- Recent Addresses (Extracted from Mail emails: "From …")
  - OS X:
    `~/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentsd-com.apple.corerecents.map-locations.plist`
  - iOS:
    `/private/var/mobile/Library/SyncedPreferences/com.apple.cloudrecents.CloudRecentsAgent-com.apple.corerecents.map-locations.plist`
- Recent Locations and Searches
- OS X:
  - `~/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.history.plist`
  - `/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.history.plist`
- iOS:
  - `/private/var/mobile/Library/SyncedPreferences/com.apple.Maps-com.apple.Maps.recents.plist`
  - `/private/var/mobile/Containers/Data/Application/<GUID>/Library/SyncedPreferences/com.apple.Maps-com.apple.Maps.recents.plist`
  - `/private/var/mobile/Containers/Data/Application/<GUID>/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.history.plist`

Synced Preferences: Maps—Recent Addresses

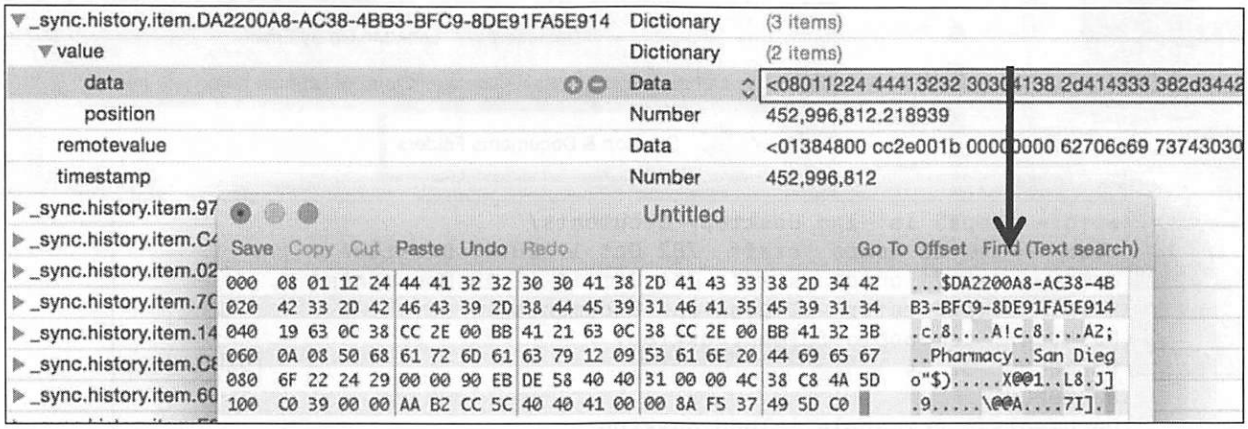
- Extracted from Mail emails: "From …"

▼ MR_F8EA2DAFA7D6877F5B232D00534B0797558A5D6C	Dictionary	(3 items)
▼ value	Dictionary	(7 items)
k	String	m
▼ t	Array	(4 items)
Item 0	Date	Apr 26, 2015, 9:58:31 AM
Item 1	Date	Apr 26, 2015, 7:25:33 AM
Item 2	Date	Apr 15, 2015, 8:26:02 PM
Item 3	Date	Apr 15, 2015, 7:35:59 PM
▼ m	Dictionary	(4 items)
▼ corerecents:from	Dictionary	(3 items)
kind	String	email
address	String	listings@redfin.com
displayName	String	Redfin
corerecents:event-time	Date	Apr 26, 2015, 7:23:45 AM
corerecents:reference-url	String	message:%3Cdata-listingAlerts-20150424_1041_740ed27991d0e463fd3fc
corerecents:subject	String	Status change on 5035 25TH St SOUTH; New Hot Home on 3103 19TH St
S	String	com.apple.mobilemail
v	Number	1
a	String	5035 25TH St SOUTH
w	Number	5
remotevalue	Data	<01384800 8726ed1a 00000000 62706c69 73743030 d7010203 04050607
timestamp	Number	451,749,511

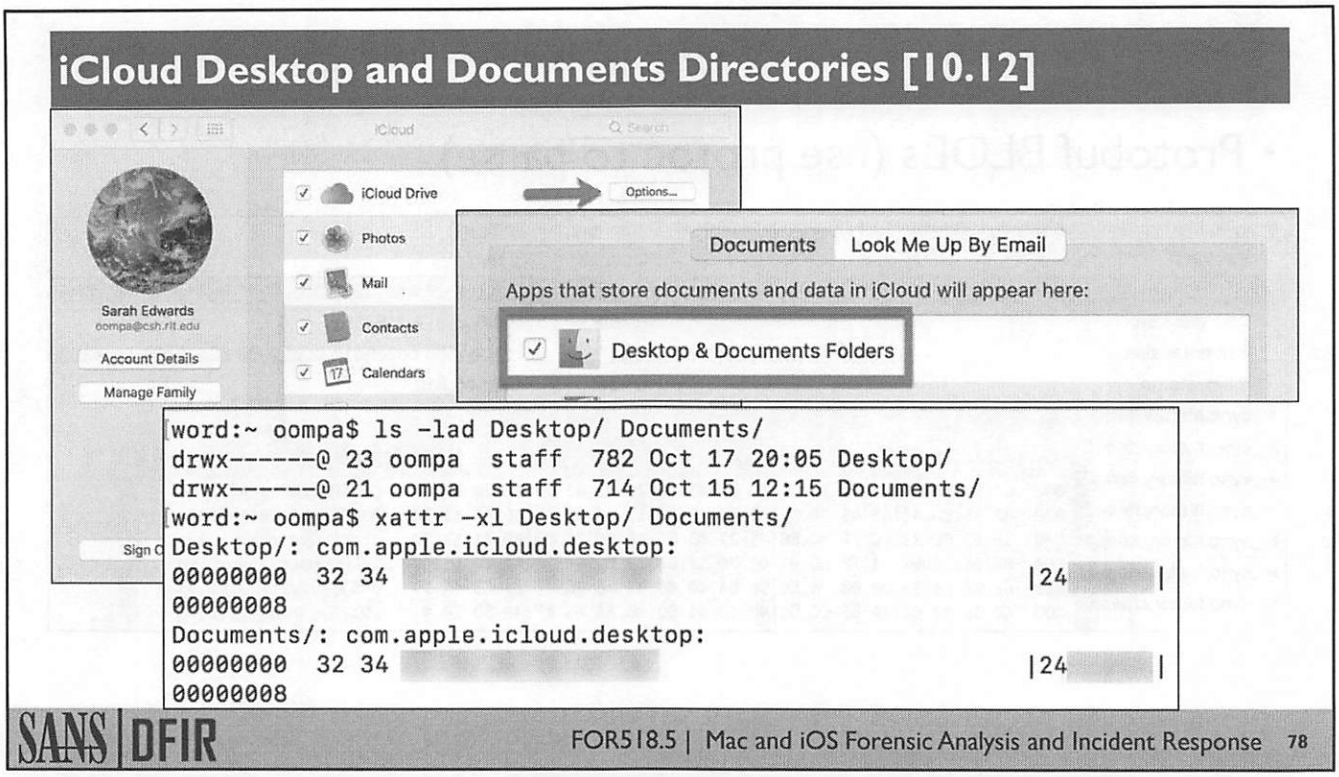
This page intentionally left blank.

Synced Preferences: Maps—Recent Locations and Searches

- Protobuf BLOBs (use protoc to parse)

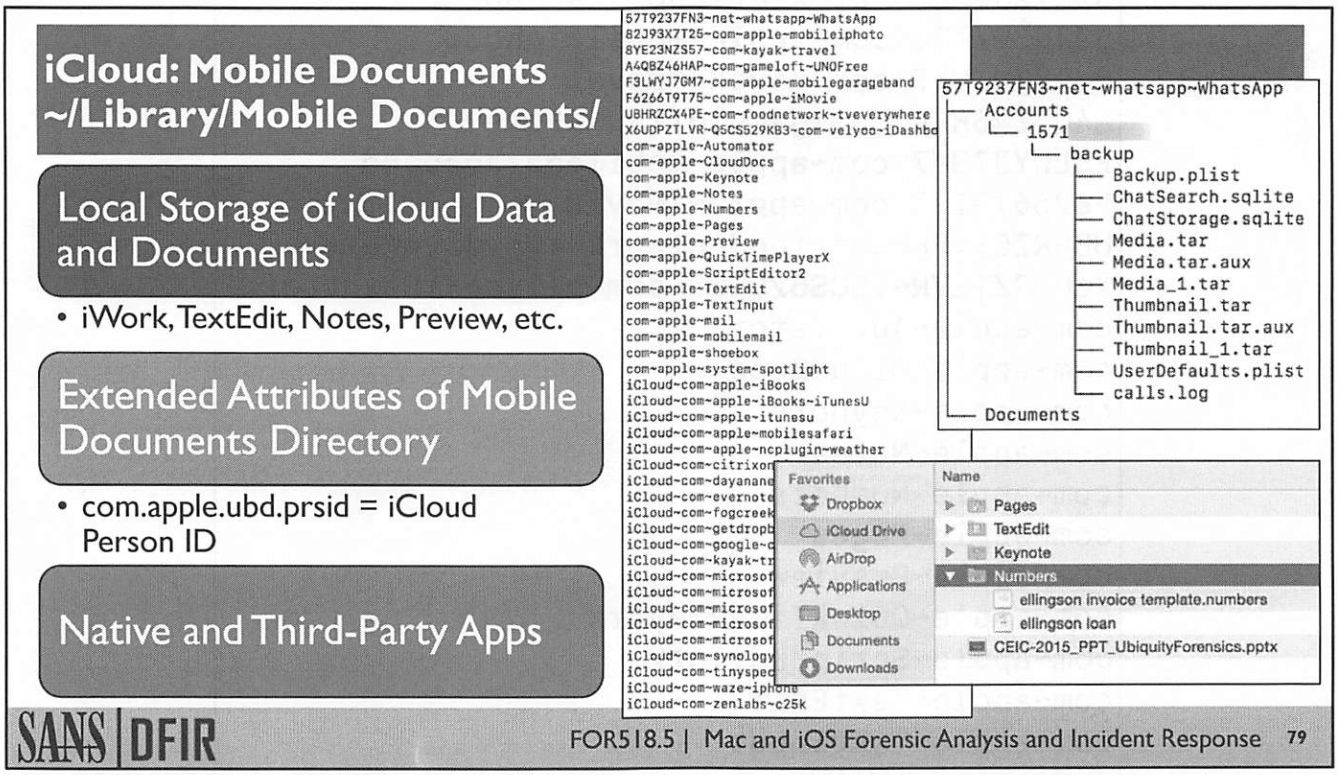


This page intentionally left blank.



With Sierra (10.12), the user now has the option to store files from their `~/Desktop` and `~/Documents` directories in their iCloud Drive—meaning these files will be synced and accessible with all devices that have access to their iCloud account. This option can be selected in the iCloud preferences panel under "Options" for iCloud Drive.

From a forensic standpoint, this can be viewed and determined through extended attributes of these directories. If the user has this feature enabled, it will show the attribute "`com.apple.icloud.desktop`" with the value of their iCloud Person ID.

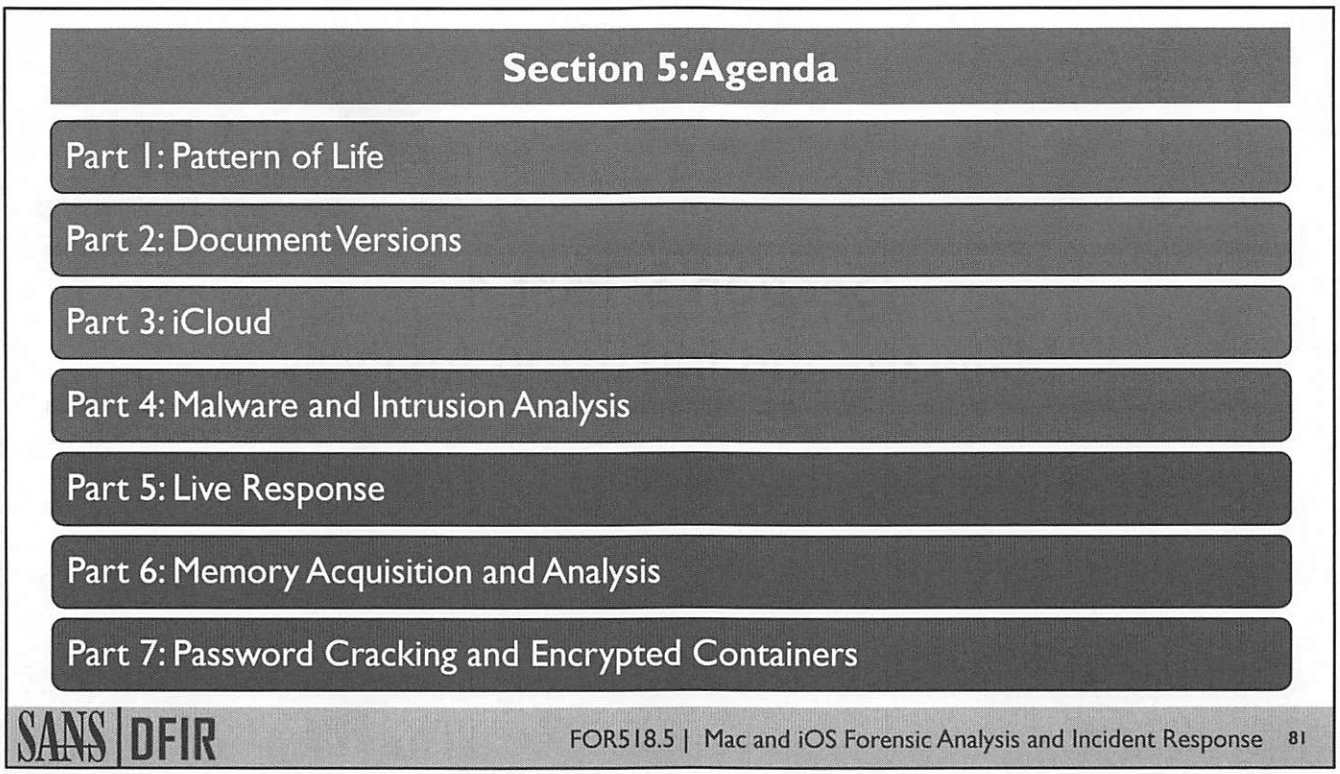


iCloud has the ability to store documents from a variety of applications, including iWork, TextEdit, and Notes. Copies of these files are stored on an OS X system in the `~/Library/Mobile Documents/` directory.

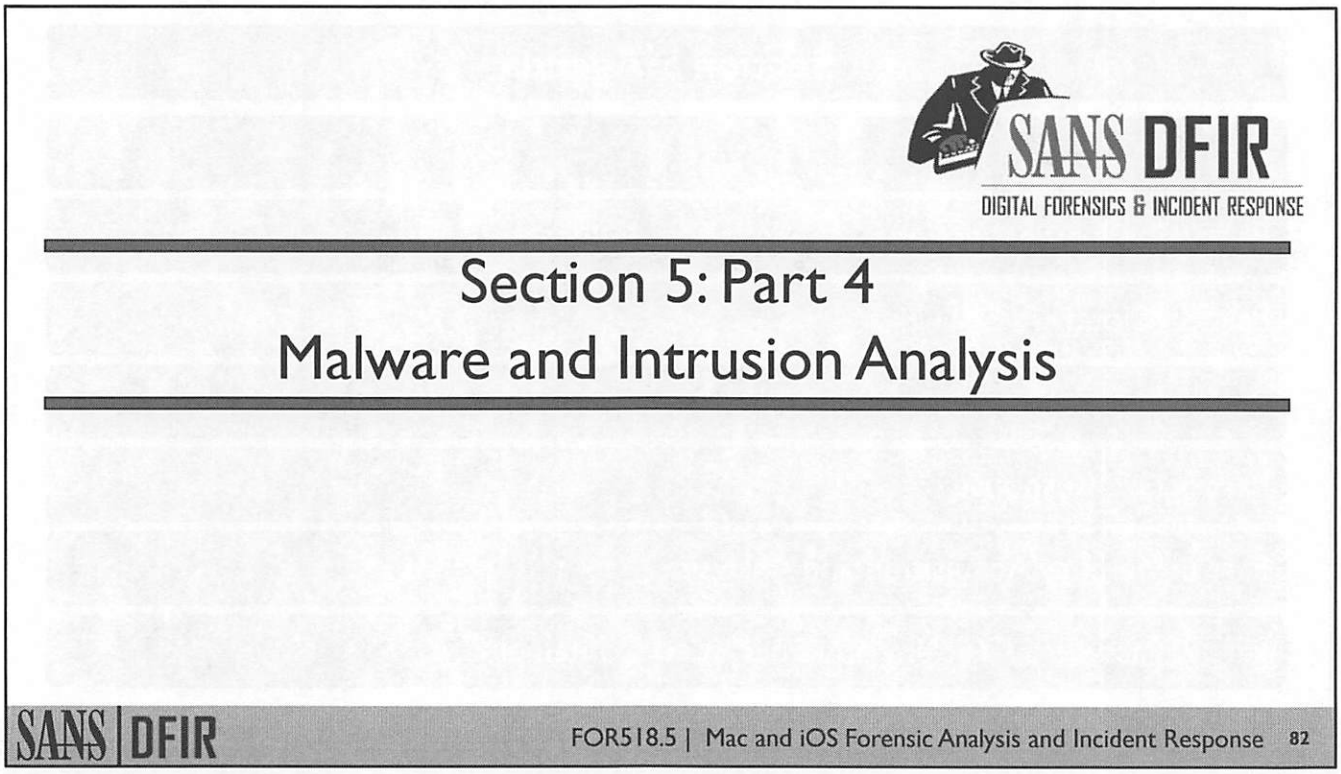
The screenshot shows a tree command output of the `Mobile Documents` directory. Each application has its own directory named with the reverse DNS naming scheme as shown before, but with tildes (~) instead of periods. For example, the TextEdit documents are stored in a directory named `com~apple~TextEdit`. The documents will be stored in the `Documents` subdirectory. iWork applications (Pages, Numbers, Keynote) contain another subdirectory, `iWorkPreviews`, that contains JPG document previews of each document.

Another way to determine an iCloud Person ID from these documents is to view their extended attributes. The attribute `com.apple.ubd.prsid` contains this numeric identifier.

```
57T9237FN3~net~whatsapp~WhatsApp
82J93X7T25~com~apple~mobileiphoto
8YE23NZS57~com~kayak~travel
A4QBZ46HAP~com~gameloft~UNOFree
F3LWYJ7GM7~com~apple~mobilegarageband
F6266T9T75~com~apple~iMovie
UBHRZCX4PE~com~foodnetwork~tveverywhere
X6UDPZTLVR~Q5CS529KB3~com~velyoo~iDashboard
com~apple~Automator
com~apple~CloudDocs
com~apple~Keynote
com~apple~Notes
com~apple~Numbers
com~apple~Pages
com~apple~Preview
com~apple~QuickTimePlayerX
com~apple~ScriptEditor2
com~apple~TextEdit
com~apple~TextInput
com~apple~mail
com~apple~mobilemail
com~apple~shoebox
com~apple~system~spotlight
iCloud~com~apple~iBooks
iCloud~com~apple~iBooks~iTunesU
iCloud~com~apple~itunesu
iCloud~com~apple~mobilesafari
iCloud~com~apple~ncplugin~weather
iCloud~com~citrixonline~iOS~GoToMeeting
iCloud~com~dayananetworks~myaltitude
iCloud~com~evernote~iPhone~Evernote
iCloud~com~fogcreek~trello
iCloud~com~getdropbox~Dropbox
iCloud~com~google~container
iCloud~com~kayak~travel
iCloud~com~microsoft~Office~Excel
iCloud~com~microsoft~Office~PowerPoint
iCloud~com~microsoft~Office~Word
iCloud~com~microsoft~onenote
iCloud~com~microsoft~skydrive
iCloud~com~synology~DSfile
iCloud~com~tinyspeck~chatlyio
iCloud~com~waze~iphone
iCloud~com~zenlabs~c25k
```



This page intentionally left blank.



This page intentionally left blank.

Apple Malware

"Macs ~~don't~~ do get hacked!"

Current Mac Malware Trends

- User-initiated via social engineering
- Java vulnerabilities
- Multi-platform malware
- Many target NGOs

Intrusion and Malware Analysis

- Similar processes and techniques
- Different files and (some) different tools

Macs do in fact get hacked (every once in a while). Their popularity is opening up a whole new market for attackers. Java vulnerabilities in particular have created a market that is creating multi-platform malware. Why target just Windows users when everyone who uses Java is vulnerable?

Mac malware on the whole is not as advanced as Windows malware, but frankly, it still gets the job done! A user will click on just about anything, given the right phishing email. Most of the Mac malware currently out there needs to be initiated by the user, so the vector tends to be an attachment via an email or a link to a malicious website.

Analysis of a Mac malware intrusion is very similar to that of a Windows intrusion—the techniques and processes are similar, but the files and tools may be different.

macOS Malware: Flashback

- Infected 600,000+ systems
- $10,000/day ad-click revenue for attackers
- Java vulnerabilities
- Fake Adobe Flash installer
- Drive-by-Download via compromised WordPress blogs

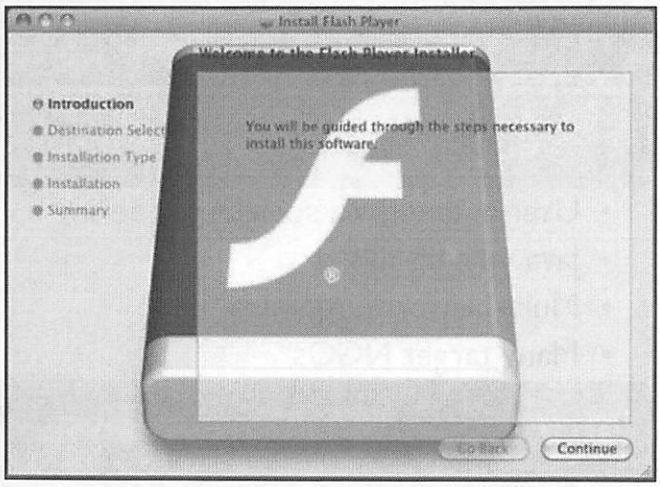


Image Source: http://www.cultofmac.com/124840/new-flashback-os-x-trojan-is-in-the-wild-and-it-can-kill-os-xs-anti-malware-scams/

The Flashback malware infected more than 600,000 systems. In terms of Mac infections, this was HUGE! The Windows CodeRed infection rate was ~400,000. (Compared to current Windows infections, it's a drop in the bucket—but you get my point).

This malware was used to generate ad-click revenue for the attackers, and was delivered by a Java vulnerability that showed up as a fake Adobe Flash installer.

macOS Malware: CoinThief/StealthBit

- Installed browser extensions in Safari and Chrome
- "Pop-Up Blocker"
- Snoops browser traffic for Bitcoin credentials (and other interesting data)
- Sends data to C2 Server

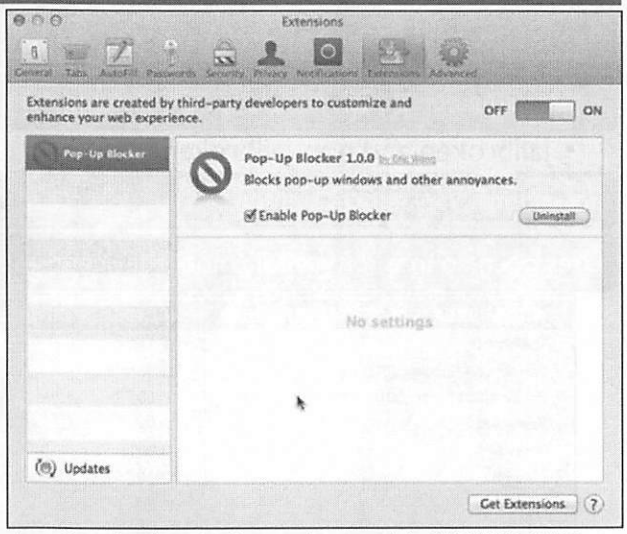


Image Source: http://www.thesafemac.com/wp-content/uploads/2014/02/CoinThief-extension.png

The CoinThief/StealthBit malware installed a browser extension to capture credentials to specific Bitcoin-related websites. These credentials were then sent to a command and control (C2) server.

References:
`http://readwrite.com/2014/02/10/stealthbit-mac-osx-trojan-malware-steals-bitcoins`
`https://www.securemac.com/privacyscan/new-apple-mac-trojan-called-osxcointhief-discovered`

macOS/iOS Malware: Wirelurker

Repackaged and trojanized third-party OS X applications on Maiyadi App Store (Chinese)

Infects connected iOS devices via OS X using dynamically generated malicious apps

- Jailbroken and non-jailbroken

Persistence via LaunchDaemon

Uses open-source software libimobiledevice to monitor for USB connections

WIRELURKER INFECTED APPLICATION	NUMBER OF DOWNLOADS
The Sims 3	42,110
International Snooker 2012	22,353
Pro Evolution Soccer 2014	20,800
Bejeweled 3	19,016
Angry Birds	14,009
Spider 3	12,745
NBA 2K13	11,113
GRID	10,820
Battlefield: Bad Company 2	8,065
Two Worlds II Game of the Year Edition	6,451

Image Source:
https://www.paloaltonetworks.com/content/dam/paloaltonetworks-com/en_US/assets/pdf/reports/Unit_42/unit42-wirelurker.pdf

Wirelurker is significant because it uses infected OS X systems to further infect iOS devices. Both jailbroken and non-jailbroken methods are used to infect the iDevices. OS X systems are initially infected by users downloading trojanized third-party applications through Chinese app stores.

Reference:

`https://www.paloaltonetworks.com/content/dam/pan/en_US/assets/pdf/reports/Unit_42/unit42-wirelurker.pdf`

macOS Malware: Crisis/Morcut

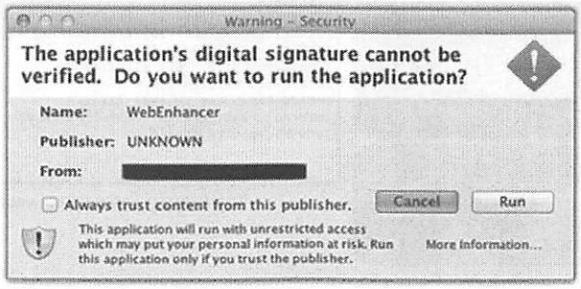
Rootkit and Spyware

Arrives as AdobeFlashPlayer.jar

- WebEnhancer.class

Cross-platform (Windows!)

Backdoor Access: Screenshots, keylog, webcam, location, microphone, files, IM data, etc.



http://nakedsecurity.sophos.com/2012/07/25/mac-malware-crisis-on-mountain-lion-eve/

The Crisis/Morcut malware is interesting because it is cross-platform. This malware arrives via Java as another fake Adobe Flash Player software.

This software creates backdoor access on the system and is reported to get access to take screenshots, install a keylogger, and access the webcam, locational data, microphone, files, and instant messenger data.

One of the malicious Java `.class` files for this malware is named `WebEnhancer.class`.

References:

`http://nakedsecurity.sophos.com/2012/07/25/mac-malware-crisis-on-mountain-lion-eve/`
`http://nakedsecurity.sophos.com/2012/07/26/mac-malware-spies-morcut-crisis/`
`http://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/OSX~Morcut-A/detailed-analysis.aspx`
`http://www.intego.com/mac-security-blog/new-apple-mac-trojan-called-osxcrisis-discovered-by-intego-virus-team/`
`http://www.symantec.com/connect/blogs/crisis-windows-sneaks-virtual-machines`

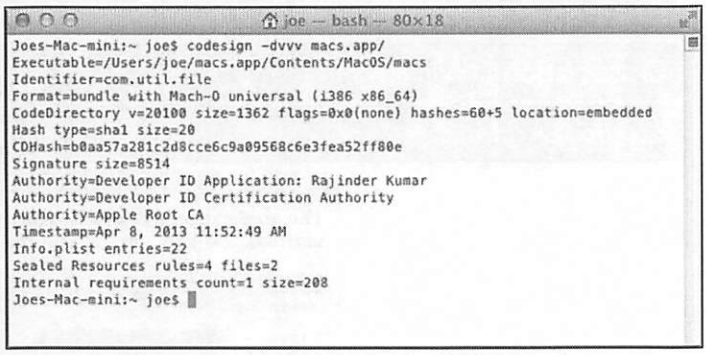
macOS Malware: KitM (Kumar-in-the-Mac)

Found on Angolan activist's system at Oslo Freedom Forum

Backdoor

Takes periodic screenshots

Signed with Apple Developer ID



http://www.f-secure.com/weblog/archives/00002554.html

The Kumar-in-the-Mac, or KitM, malware was unique because it was the first major piece of malware that was found signed by a legitimate Apple Developer ID.

This malware was found on another activist system at the Oslo Freedom Forum in Norway. The malware has simple backdoor access and has the ability to take periodic screenshots of the system.

macOS Malware: KeRanger

- Ransomware: Encrypted files—User must pay with Bitcoin
- Infected legitimate downloads of torrent client Transmission
- Signed with Developer ID
- UPX Packed

README_FOR_DECRYPT.txt — Edited

```
Your computer has been locked and all your files has been encrypted with 2048-bit RSA encryption.

Instruction for decrypt:

1. Go to https://fiwf4kwysm4dpw5l.onion.to ( IF NOT WORKING JUST DOWNLOAD TOR BROWSER AND OPEN
THIS LINK: http://fiwf4kwysm4dpw5l.onion )
2. Use 1PGAUBqHNcwSHYKnpHqzCrPkyxNxvsmEof as your ID for authentication
3. Pay 1 BTC (~407.47$) for decryption pack using bitcoins (wallet is your ID for authentication -
1PGAUBqHNcwSHYKnpHqzCrPkyxNxvsmEof)
4. Download decrypt pack and run


----> Also at https://fiwf4kwysm4dpw5l.onion.to  you can decrypt 1 file for FREE to make sure
decryption is working.


Also we have ticket system inside, so if you have any questions - you are  welcome.
We will answer only if you able to pay and you have serious question.

IMPORTANT: WE ARE ACCEPT ONLY(!!) BITCOINS
HOW TO BUY BITCOINS:
https://localbitcoins.com/guides/how-to-buy-bitcoins
https://en.bitcoin.it/wiki/Buying_Bitcoins_(the_newbie_version)
```

http://researchcenter.paloaltonetworks.com/2016/03/new-os-x-ransomware-keranger-infected-transmission-bittorrent-client-installer/

Windows systems have had ransomware malware for a few years now. OS X just got its first one! This piece of malware was acquired by a legitimate download of the popular torrent client, Transmission.

This malware will encrypt the user's files, and they will have to pay Bitcoin to get them decrypted. This is also one of the first pieces of Mac malware that implements a packer.

Reference:
`https://unit42.paloaltonetworks.com/new-os-x-ransomware-keranger-infected-transmission-bittorrent-client-installer/`

iOS Malware: Pegasus Spyware

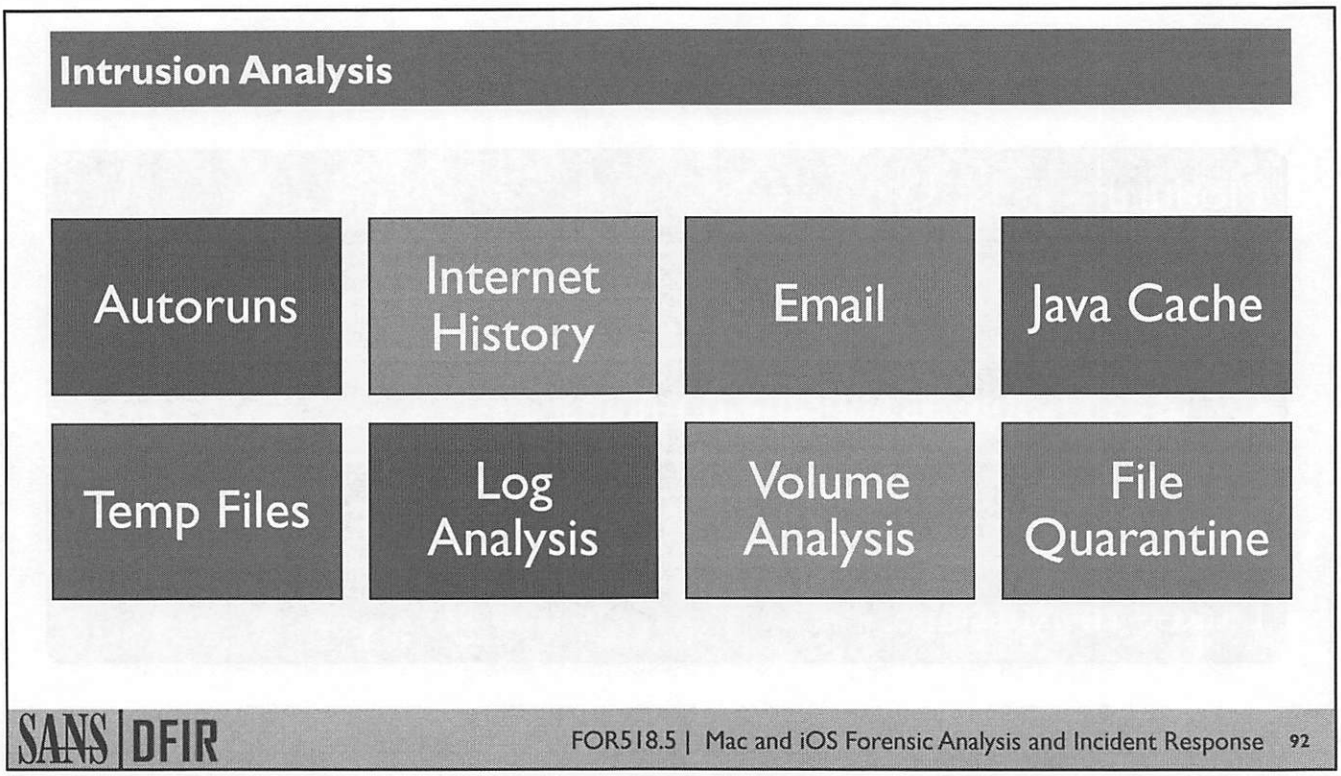
- iOS Malware
- Sent Link via Text Message to Human Rights Defender, Ahmed Mansoor
- NSO Group: Israel-based
- "Enterprise Class" Malware: Lookout
- Trident Vulnerabilities: Three Zero-Day Vulnerabilities
- Patched in 9.3.5

Reference:
`https://blog.lookout.com/trident-pegasus`

macOS Malware: Sofacy | APT28 | Sednit APT: Xagent Payload

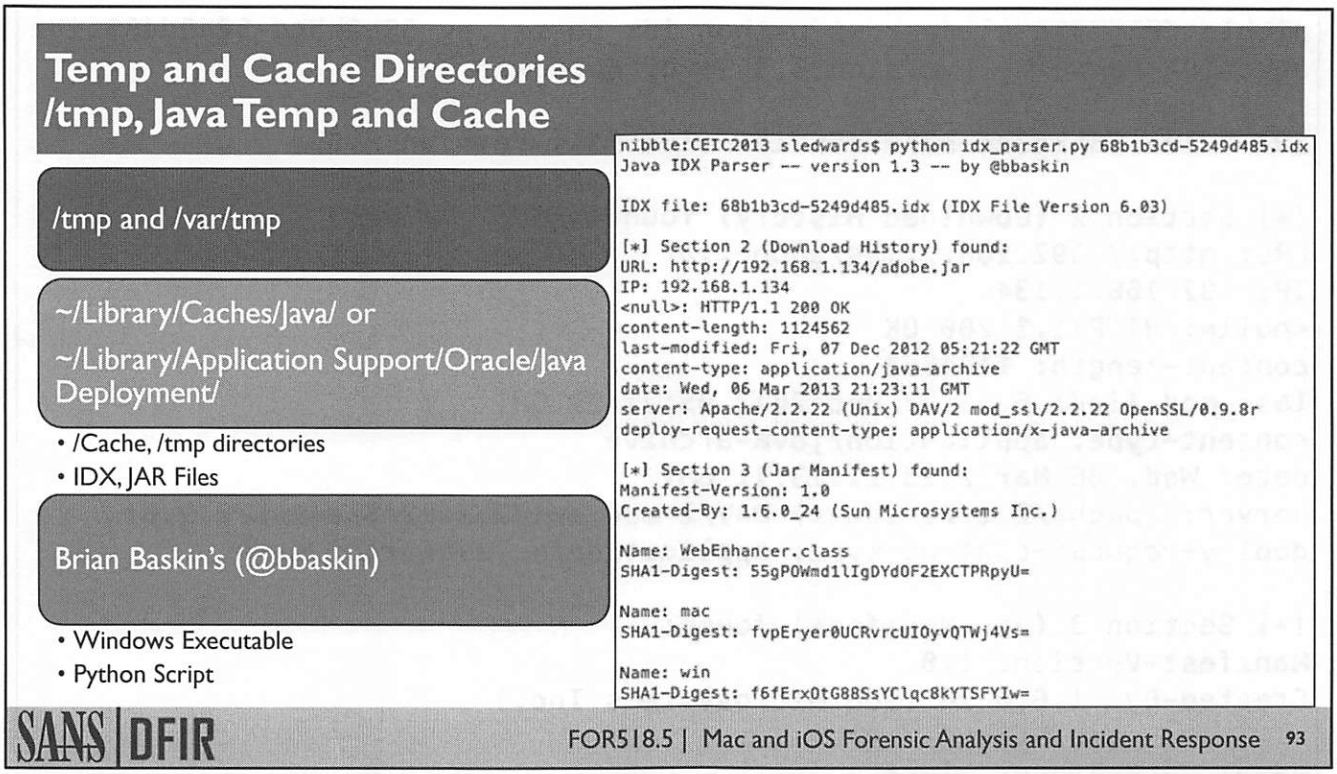
- Modular
- Acquires iOS backups, passwords, screenshots, etc.
- C2 uses "Apple"-sounding domains
- Shares artifacts with Komplex OS X Malware Downloader
- Linked to Russia

Reference:
https://labs.bitdefender.com/2017/02/new-xagent-mac-malware-linked-with-the-apt28/



SANS FOR508 does a really great job of walking an investigator through a Windows intrusion analysis. We can use these same processes and techniques on a Mac intrusion analysis. The only difference is the file types and tools used.

While most of these topics have already been covered in the class, such as Mac autoruns, internet and email, log analysis, and volume analysis, there are a few more areas specific to finding Mac malware on a system. These topics include temporary directories, Java cache files, and the file quarantine process.



Temporary directories such as `/tmp` and `/var/tmp` have a tendency to be locations where malware files or decoy documents are written when malware is executed on the system, likely because they have permissions to write in these locations.

Java cache files located in the `~/Library/Caches/Java/` directory may contain Java IDX and JAR files. These files may need further investigation to determine if a Java-based piece of malware was executed on the system. Java ARchive (JAR) files, are the binary that is executed on the system. The IDX files are a Java index file that shows where the JAR file came from and other metadata.

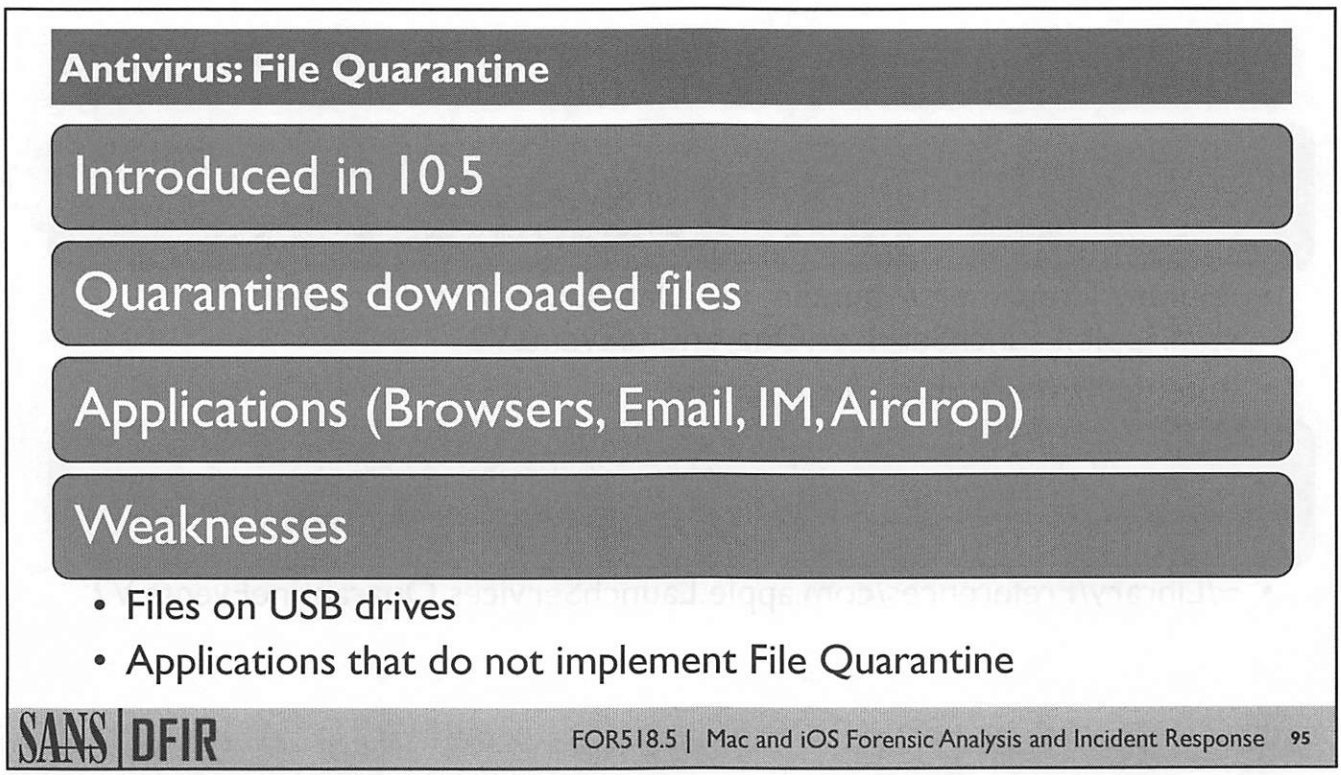
Java IDX files contain data such as the IP or web address showing where the Java JAR file was downloaded from, when it was downloaded, how large it is, file checksums, and the contents of the Java archive. Only some of the data can be parsed by the human eye in the hex view of the file.

Brian Baskin (`@bbaskin`) developed a great IDX file parser that can be used on Windows platforms with a Windows executable, or on most other platforms with a Python script.

The screenshot shows an example of an IDX file that contains information for the JAR file related to the Crisis malware. The JAR file named `adobe.jar` was downloaded from `http://192.168.1.134` on March 6, 2013. This JAR file contains the `WebEnhancer.class` file as well as cross-platform binaries named "`mac`" and "`win`".

Reference:
`http://github.com/Rurik/Java_IDX_Parser`

```
nibble:CEIC2013 sledwards$ python idx_parser.py 68b1b3cd-5249d485.idx
Java IDX Parser -- version 1.3 -- by @bbaskin

IDX file: 68b1b3cd-5249d485.idx (IDX File Version 6.03)

[*] Section 2 (Download History) found:
URL: http://192.168.1.134/adobe.jar
IP: 192.168.1.134
<null>: HTTP/1.1 200 OK
content-length: 1124562
last-modified: Fri, 07 Dec 2012 05:21:22 GMT
content-type: application/java-archive
date: Wed, 06 Mar 2013 21:23:11 GMT
server: Apache/2.2.22 (Unix) DAV/2 mod_ssl/2.2.22 OpenSSL/0.9.8r
deploy-request-content-type: application/x-java-archive

[*] Section 3 (Jar Manifest) found:
Manifest-Version: 1.0
Created-By: 1.6.0_24 (Sun Microsystems Inc.)

Name: WebEnhancer.class
SHA1-Digest: 55gP0Wmd1lIgDYdOF2EXCTPRpyU=

Name: mac
SHA1-Digest: fvpEryer0UCRvrcUIOyvQTWj4Vs=

Name: win
SHA1-Digest: f6fErxOtG88SsYClqc8kYTSFYIw=
```



Apple introduced the concept of file quarantining in 10.5. It is a method that OS X uses to tag a file with where it came from so it can be checked by the Xprotect antivirus solution.

Files get quarantined if they are downloaded by applications that implement this feature. This can be checked by looking at the `LSFileQuarantineEnabled` key in an Applications `Info.plist` file in the Application bundle. If it is implemented, it will be set to "`True`". Most popular web browsers and email clients implement this on the Mac today.

The File Quarantine method does have weaknesses. For instance, it does not quarantine files that are copied off of thumb drives or from applications that do not implement the File Quarantine functionality.

Antivirus: File Quarantine Events SQLite Database

10.11+

- /Library/Containers/<bundle_id>/Data/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2
- Potentially more than one database!

10.7+

- ~/Library/Preferences/com.apple.LaunchServices.QuarantineEvents.V2

A SQLite database containing file quarantine data is located in the user's `~/Library/Preferences (legacy or sandboxed)` directory. On systems 10.7 and above, it is named `com.apple.LaunchServices.QuarantineEvents.V2`. On macOS 10.11 and newer, this database can be found in multiple places—make sure to look for all of them! Each may have separate quarantine entries for different applications.

Antivirus: File Quarantine Event Example

- Quarantine Events: LSQuarantineEvent Table

Database Column	Example Data
LSQuarantineEventIdentifier	68F08939-EF7F-4326-BDA3-810542E43579
LSQuarantineTimeStamp	358820762.0
LSQuarantineAgentBundleIdentifier	com.google.Chrome
LSQuarantineAgentName	Google Chrome
LSQuarantineDataURLString	http://ash.barebones.com/TextWrangler_4.0.dmg
LSQuarantineSenderName	*NULL*
LSQuarantineSenderAddress	*NULL*
LSQuarantineTypeNumber	0
LSQuarantineOriginTitle	*NULL*
LSQuarantineOriginURLString	http://www.barebones.com/products/textwrangler/
LSQuarantineOriginAlias	*NULL*

An example of a record in the File Quarantine Events database is shown above. This record has been extracted from the database and placed into this table for easier reading.

The information in the database can help an analyst determine where a certain file was downloaded from with related information.

- `LSQuarantineEventIdentifier`: Unique Event Identifier GUID
- `LSQuarantineTimeStamp`: Timestamp when file was quarantined (Mac Absolute Time/WebKit time, seconds from 1/1/2001)
- `LSQuarantineAgentBundleIdentifier`: Bundle ID of the application that downloaded the file
- `LSQuarantineAgentName`: Application that downloaded the file (e.g., Safari, Google Chrome, Mail)
- `LSQuarantineDataURLString`: URL the file was actually downloaded from (may be different from the URL used by the user to download the file)
- `LSQuarantineSenderName`: Files downloaded from the Mail applications include the email sender's name. Those downloaded from Airdrop will have the hostname sender system
- `LSQuarantineSenderAddress`: Used in files downloaded from Mail application and Email Sender's Email Address
- `LSQuarantineTypeNumber`: Quarantine Type
  - 0: Web Browsers
  - 1: Xcode
  - 2: Apple Mail
  - 3: iChat
  - 6: Airdrop/sharingd
  - 7: Other Apps (Slack App)
- `LSQuarantineOriginTitle`: Used in files downloaded from Mail and Email Subject
- `LSQuarantineOriginURLString`: URL the user visited to download the file (Browser); Email Server Information
- `LSQuarantineOriginAlias`: Unknown

Files sent via Airdrop may have the Agent Name "`NetworkBrowserAgent`".

Antivirus: XProtect

/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/ or
/System/Library/CoreServices/XProtect.bundle/Contents/Resources/

- XProtect.meta.plist
  - Last update date and version (10.8/10.7)
  - Java minimum version and blacklisted plugins
- XProtect.plist: AV signatures
- Xprotect.yara: YARA signatures

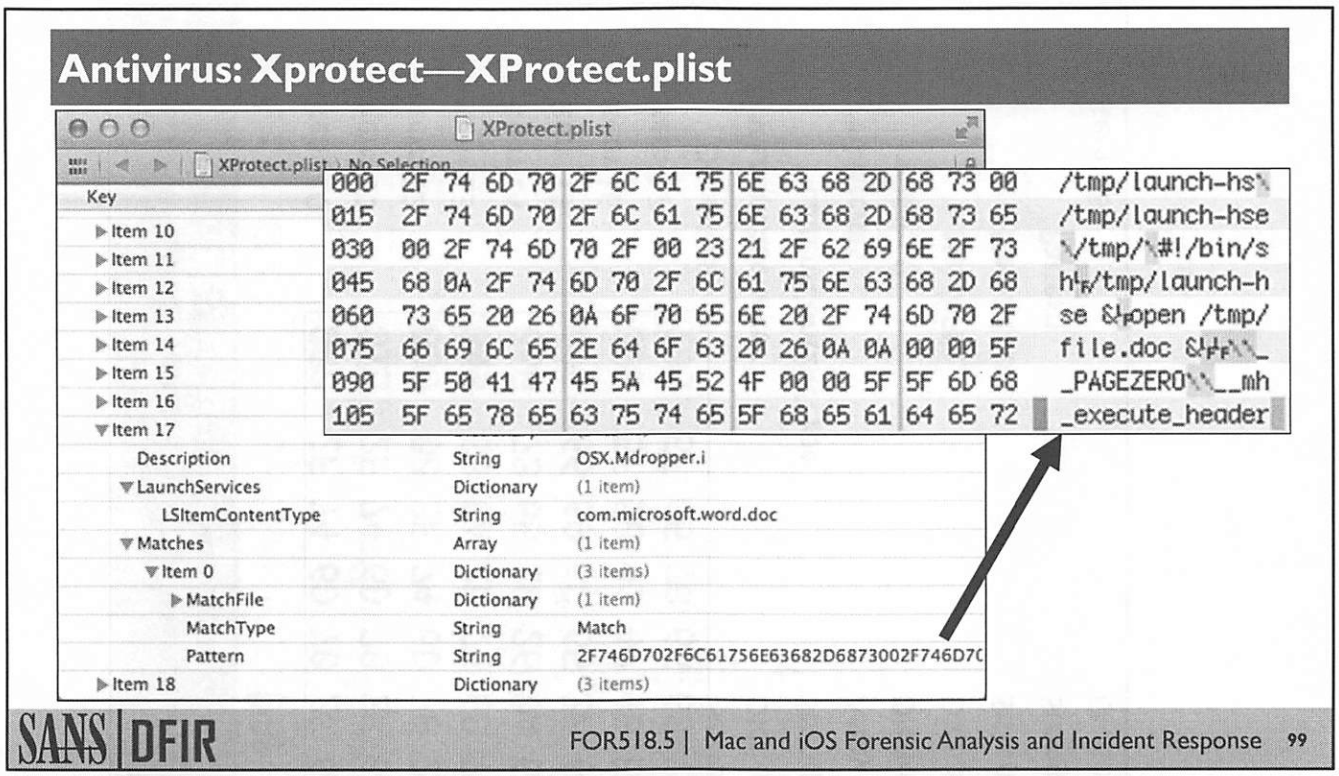
Weaknesses

- Apple updates it sometimes
- Very few signatures on blacklist
- No heuristics
- Only checks "quarantined" files
- (Mostly) Mac threats

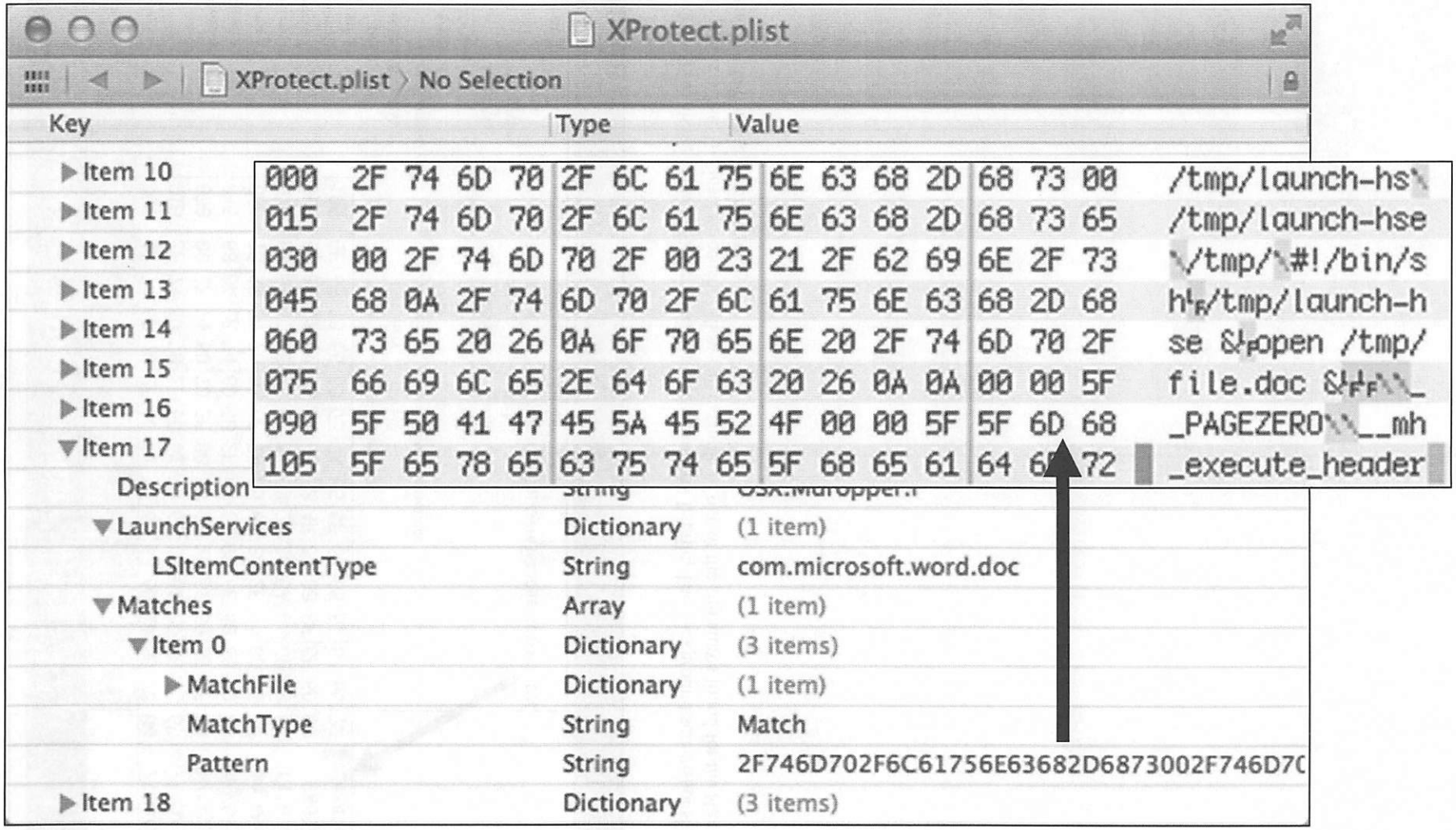
XProtect was introduced in 10.6. The XProtect system is Apple's answer to antivirus. Xprotect uses the `Xprotect.plist` property list located in the `/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/` or `/System/Library/CoreServices/XProtect.bundle/Contents/Resources/` directories. This property list contains signatures of well-known threats for the Mac. The `XProtect.meta.plist` file located in the same directory contains the date when this signature property list was last updated.

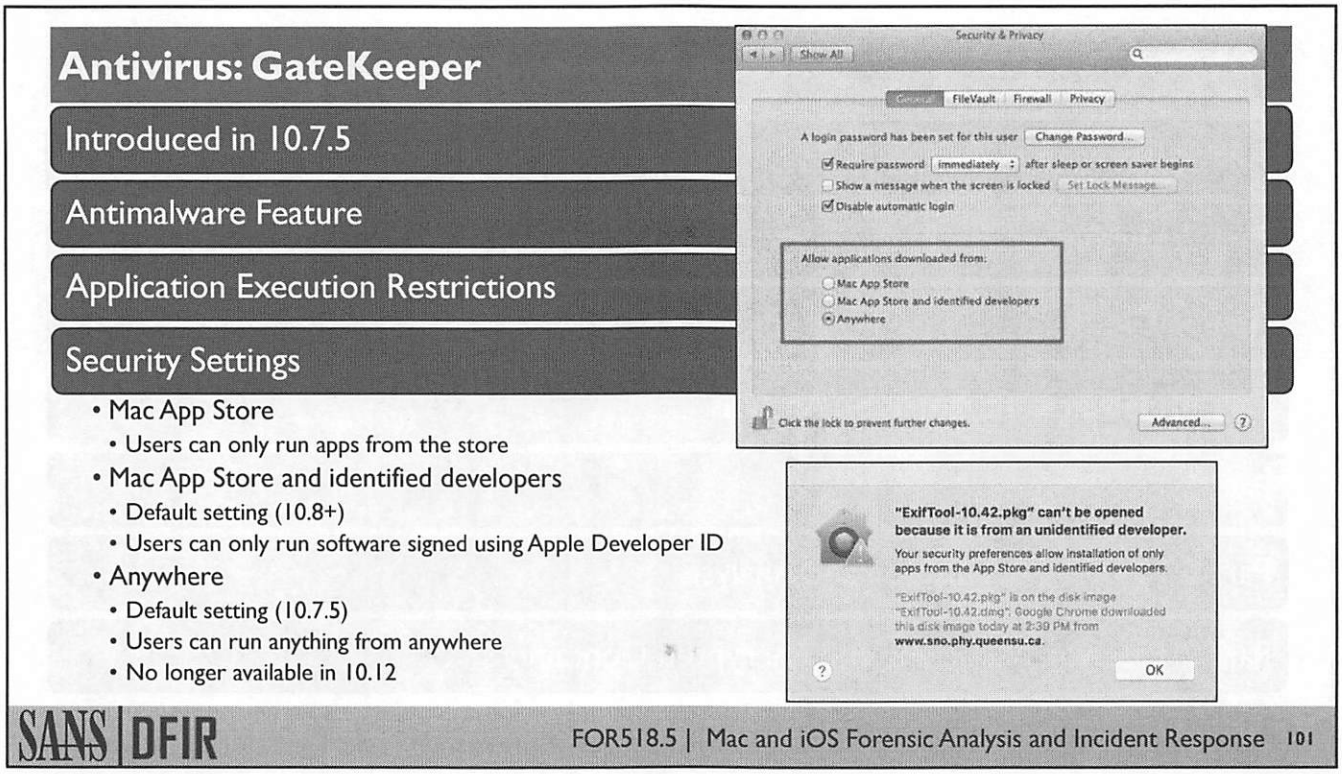
The XProtect system does have its weaknesses. For instance, it only gets updated when Apple decides to update it. It is not meant to be updated by the user. The signatures are limited, not only in quantity, but in quality. These signatures do not use heuristics and only serve to protect from Mac-related threats, some Windows based threats were found in these files too (https://twitter.com/patrickwardle/status/1120771284286103552). Most Windows antivirus products will scan for threats to other operating systems. The XProtect system only checks those files that are file quarantined. Other files transferred from USB drives or from network shares will not be checked. An example of the YARA rules is shown below.

```
rule CoinThiefC
{
    meta:
        description = "OSX.CoinThief.C"
        xprotect_rule = true
    condition:
        Macho and filesize <= 29000 and
        (
            hash.sha1(0, filesize) == "d4d1480a623378202517cf86efc4ec27f3232f0d"
        )
}

rule RSPlugA
{
    meta:
        description = "OSX.RSPlug.A"
        xprotect_rule = true
    strings:
        $a = {4D6F7A696C6C61706C75672E706C7567696E00 [-] 566572696669656444 6F776E6C6F6164506C7567696E00 [-] 56657269
66696564446F776E6C6F6164506C7567696E2E7273726300}
        $b = {3C6B65793E4946506B67466C616744656661756C744C6F636174696F6E3C2F6B65793E [-] 3C737472696E673E2F4C6962726
172792F496E7465726E657420506C75672D496E732F3C2F737472696E673E}
        $c = {23212F62696E2F [0-2] 7368}
    condition:
        $a and $b and $c
}
```



The screenshots above show the `XProtect.plist` property list file. The background screenshot shows the whole property list. The inset screenshot shows an example of one of the signatures found for the `MacControl` malware located in the `Pattern` key.





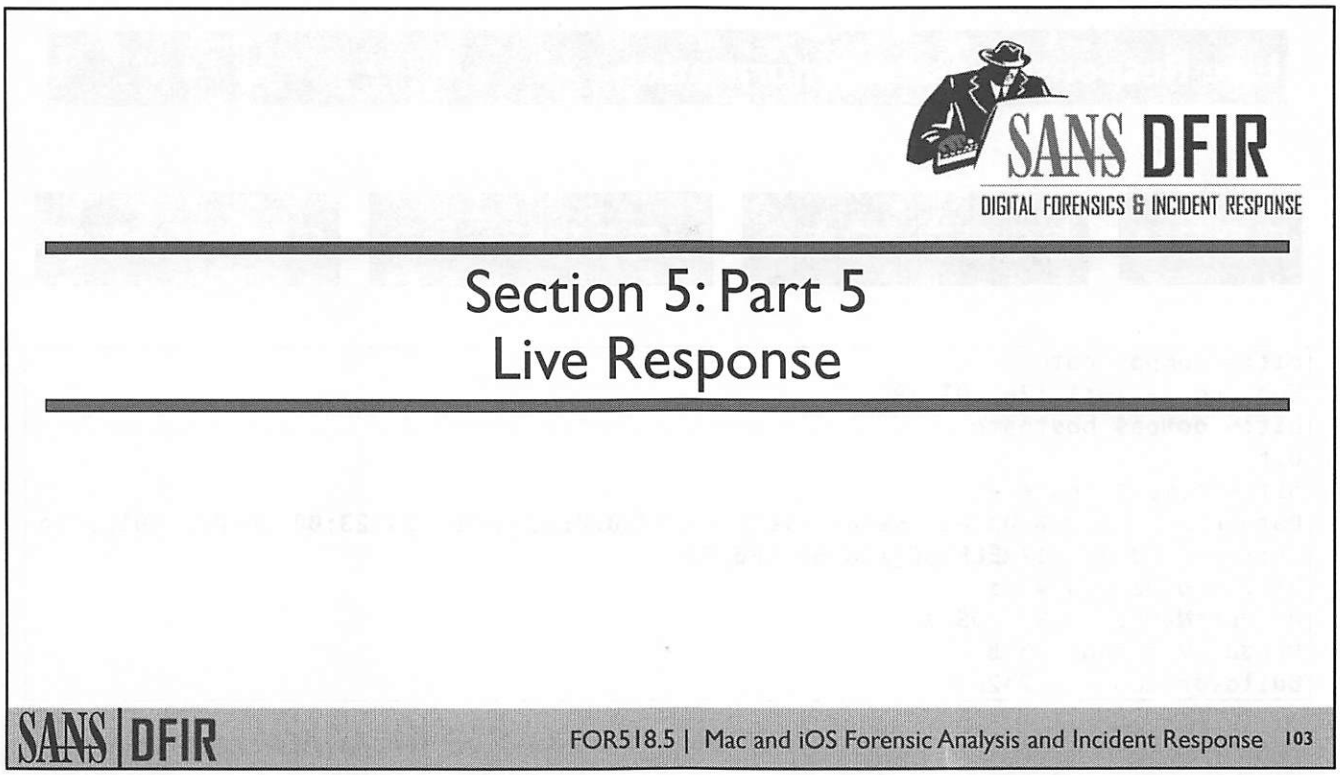
Gatekeeper was introduced in 10.7.5. This functionality allowed the user to choose the level of security used when downloading applications from the internet. Three options existed for the user to choose from.

- Mac App Store: Only allows applications to open that were downloaded from the Mac App Store.
- Mac App Store and identified developers: Default selection in 10.8+. This limits applications to open only if they came from the Mac App Store, or are signed with an Apple Developer ID.
- Anywhere: Default selection in 10.7.5. This will allow applications downloaded from anywhere to open.

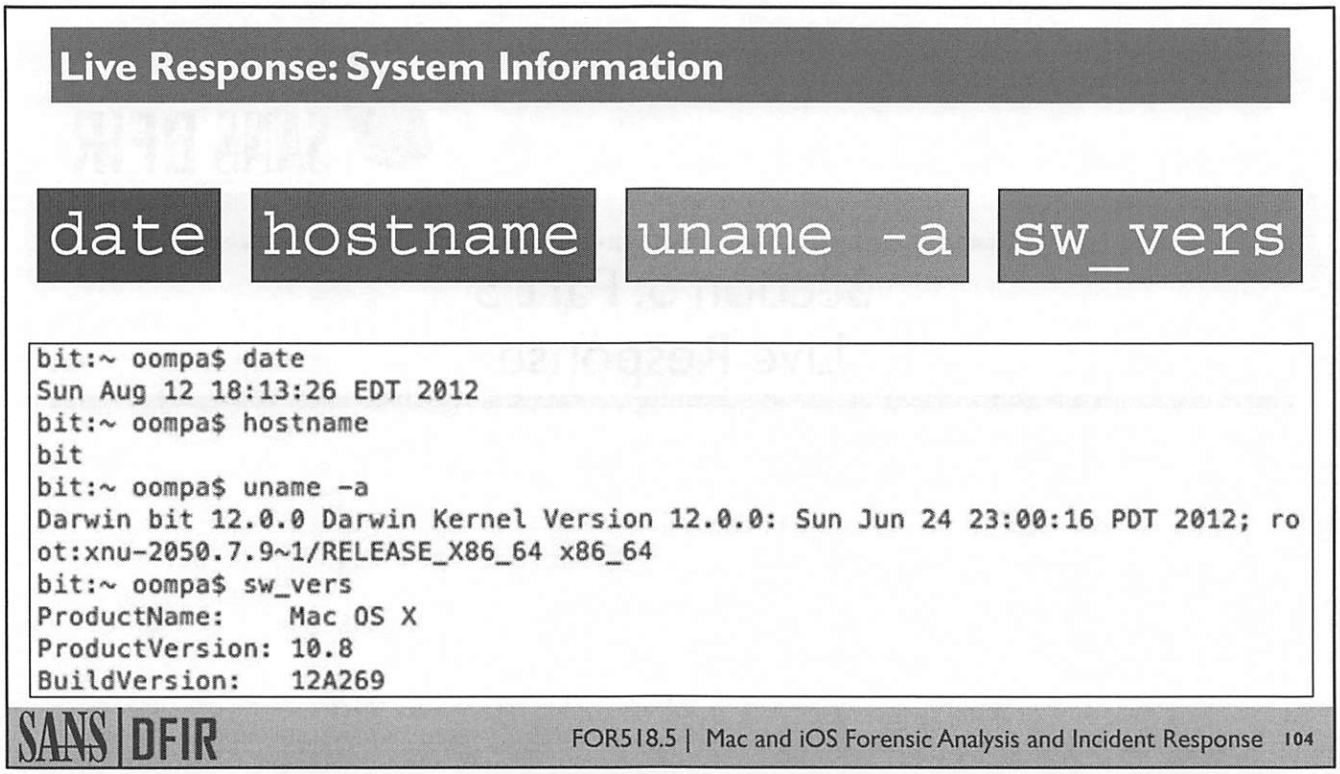
Section 5: Agenda

- Part 1: Pattern of Life
- Part 2: Document Versions
- Part 3: iCloud
- Part 4: Malware and Intrusion Analysis
- Part 5: Live Response
- Part 6: Memory Acquisition and Analysis
- Part 7: Password Cracking and Encrypted Containers

This page intentionally left blank.



This page intentionally left blank.



These four commands will give you the basic information about the system.

`date`: The local time of the system.
`hostname`: The hostname of the system.
`uname -a`: Prints the OS name, network node name, system architecture/release/name/version information. (The `-a` parameter displays all the information available for this command).
`sw_vers`: Prints the OS X version and build information.

Many similar commands in other systems do not necessarily work by default on the Mac. The Mac may use a completely different command to perform the same action, or it may just implement different option flags. Getting to know the commands and using man pages will help you understand the command-line intricacies of the Mac.

More information about mapping the Darwin Kernel versions can be found here:
`http://en.wikipedia.org/wiki/Darwin_(operating_system)`

Live Response: Active Network Connections

netstat -anf inet

```
nibble:vm sledwards$ netstat -anf inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.206.61743    74.125.228.71.443      ESTABLISHED
tcp4       0      0  192.168.1.206.61742    74.125.228.37.443      ESTABLISHED
tcp4      37      0  192.168.1.206.61735    107.20.249.221.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61734    107.20.249.221.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61733    108.160.166.10.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61732    108.160.165.253.443    CLOSE_WAIT
tcp4       0      0  192.168.1.206.61720    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61719    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61718    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61717    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61651    174.129.31.99.443      ESTABLISHED
tcp4       0      0  192.168.1.206.61650    174.129.31.99.443      ESTABLISHED
```


Was the system connected to a lewd server or a malicious IP address? The `netstat -an` command lists active network connections. These can be used to determine what connections the system is maintaining.

The `-a` parameter shows the state of all network sockets.

The `-n` parameter shows network addresses as numbers (rather than host and domain names).

The `-f` parameter limits the amount of data shown to only IPv4 and IPv6 network sockets: This does not include UNIX sockets.

The output fields:

- Protocol (TCP or UDP)
- Received Queue (bytes)
- Send Queue (bytes)
- Local IP Address and Port
- Foreign (Remote) IP Address and Port
- State of the Connection

```
nibble:vm sledwards$ netstat -anf inet
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  192.168.1.206.61743    74.125.228.71.443      ESTABLISHED
tcp4       0      0  192.168.1.206.61742    74.125.228.37.443      ESTABLISHED
tcp4      37      0  192.168.1.206.61735    107.20.249.221.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61734    107.20.249.221.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61733    108.160.166.10.443     CLOSE_WAIT
tcp4      37      0  192.168.1.206.61732    108.160.165.253.443    CLOSE_WAIT
tcp4       0      0  192.168.1.206.61720    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61719    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61718    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61717    8.18.203.20.80         ESTABLISHED
tcp4       0      0  192.168.1.206.61651    174.129.31.99.443      ESTABLISHED
tcp4       0      0  192.168.1.206.61650    174.129.31.99.443      ESTABLISHED
```

Live Response: Active Network Connections by Process

lsof -i

```
bit:~ oompa$ lsof -i | more
COMMAND    PID  USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
SystemUIS  236 oompa    8u  IPv4 0x9ed2b1a706515075      0t0  UDP *:*
NetworkBr  241 oompa    5u  IPv4 0x9ed2b1a7065179cd      0t0  UDP *:*
imagent    258 oompa   12u  IPv4 0x9ed2b1a706763535      0t0  UDP localhost:58160->localhost:58160
imagent    258 oompa   13u  IPv4 0x9ed2b1a71bc9bc65      0t0  TCP bit:61027->bos-d005b-rdr4.blue.aol.com:https (ESTABLISHED)
imagent    258 oompa   16u  IPv4 0x9ed2b1a706eeb84d      0t0  TCP bit:61011->qc-in-f125.1e100.net:5223 (ESTABLISHED)
Dropbox    279 oompa   10u  IPv4 0x9ed2b1a71bca0c65      0t0  TCP bit:60999->sjc-not18.sjc.dropbox.com:http (ESTABLISHED)
Dropbox    279 oompa   16u  IPv4 0x9ed2b1a7065151fd      0t0  UDP *:17500
Dropbox    279 oompa   19u  IPv4 0x9ed2b1a70cace20d      0t0  TCP *:17500 (LISTEN)
Dropbox    279 oompa   21u  IPv4 0x9ed2b1a710367df5      0t0  TCP bit:61139->v-client-1a.sjc.dropbox.com:https (CLOSE_WAIT)
Dropbox    279 oompa   25u  IPv4 0x9ed2b1a709419c65      0t0  TCP localhost:26164 (LISTEN)
Dropbox    279 oompa   28u  IPv4 0x9ed2b1a710259ad5      0t0  TCP bit:61035->ec2-107-22-245-91.compute-1.amazonaws.com:https
 (CLOSE_WAIT)
Dropbox    279 oompa   29u  IPv4 0x9ed2b1a706eebf85      0t0  TCP bit:61045->v-client-2b.sjc.dropbox.com:https (CLOSE_WAIT)
Mail       824 oompa   34u  IPv4 0x9ed2b1a70fed8df5      0t0  TCP bit:61017->mail.csh.rit.edu:imaps (ESTABLISHED)
```


The `lsof -i` command lists internet and network connections by process. In the screenshot we can see that the Dropbox and Mail applications had active connections.

The output fields:

- Process Name
- Process ID
- User Account
- File Descriptor (See man page for details)
- Type (IPv4, IPv6)
- Device (See man page for details)
- Size/Offset (See man page for details)
- Node (TCP or UDP)
- Connection Information and Status (See man page for details)

Other flags that could be of use are the following:

- `-n`: Show IP addresses instead of network names.
- `-P`: Show port numbers versus names.
- `-l`: Show numeric user IDs versus account names.

```
bit:~ oompa$ lsof -i | more
COMMAND    PID  USER   FD   TYPE             DEVICE SIZE/OFF NODE NAME
SystemUIS  236 oompa    8u  IPv4 0x9ed2b1a706515075      0t0  UDP *:*
NetworkBr  241 oompa    5u  IPv4 0x9ed2b1a7065179cd      0t0  UDP *:*
imagent    258 oompa   12u  IPv4 0x9ed2b1a706763535      0t0  UDP localhost:58160->localhost:58160
imagent    258 oompa   13u  IPv4 0x9ed2b1a71bc9bc65      0t0  TCP bit:61027->bos-d005b-rdr4.blue.aol.com:https (ESTABLISHED)
imagent    258 oompa   16u  IPv4 0x9ed2b1a706eeb84d      0t0  TCP bit:61011->qc-in-f125.1e100.net:5223 (ESTABLISHED)
Dropbox    279 oompa   10u  IPv4 0x9ed2b1a71bca0c65      0t0  TCP bit:60999->sjc-not18.sjc.dropbox.com:http (ESTABLISHED)
Dropbox    279 oompa   16u  IPv4 0x9ed2b1a7065151fd      0t0  UDP *:17500
Dropbox    279 oompa   19u  IPv4 0x9ed2b1a70cace20d      0t0  TCP *:17500 (LISTEN)
Dropbox    279 oompa   21u  IPv4 0x9ed2b1a710367df5      0t0  TCP bit:61139->v-client-1a.sjc.dropbox.com:https (CLOSE_WAIT)
Dropbox    279 oompa   25u  IPv4 0x9ed2b1a709419c65      0t0  TCP localhost:26164 (LISTEN)
Dropbox    279 oompa   28u  IPv4 0x9ed2b1a710259ad5      0t0  TCP bit:61035->ec2-107-22-245-91.compute-1.amazonaws.com:https
 (CLOSE_WAIT)
Dropbox    279 oompa   29u  IPv4 0x9ed2b1a706eebf85      0t0  TCP bit:61045->v-client-2b.sjc.dropbox.com:https (CLOSE_WAIT)
Mail       824 oompa   34u  IPv4 0x9ed2b1a70fed8df5      0t0  TCP bit:61017->mail.csh.rit.edu:imaps (ESTABLISHED)
```

Live Response: Routing Table

netstat -rn

```
bit:~ oompa$ netstat -rn | more
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.254      UGSc           22        0     en1
127                127.0.0.1          UCS             0        0     lo0
127.0.0.1          127.0.0.1          UH              5    23616     lo0
169.254            link#6             UCS             1        0     en1
169.254.204.125    b8:c7:5d:cc:5:80   UHLSW           0        1     en1
192.168.1          link#6             UCS             5        0     en1
192.168.1.1        c0:3f:e:8c:59:59   UHLWIi          1      104     en1    850
192.168.1.101      127.0.0.1          UHS             0        0     lo0
192.168.1.133      3c:7:54:3:65:20    UHLWIi          0     4502     en1    295
192.168.1.209      d0:23:db:72:91:10  UHLWIi          0       45     en1   1163
192.168.1.254      e0:69:95:50:4c:6   UHLWIir        23      577     en1   1188
192.168.1.255      ff:ff:ff:ff:ff:ff  UHLWbI          0        1     en1
```

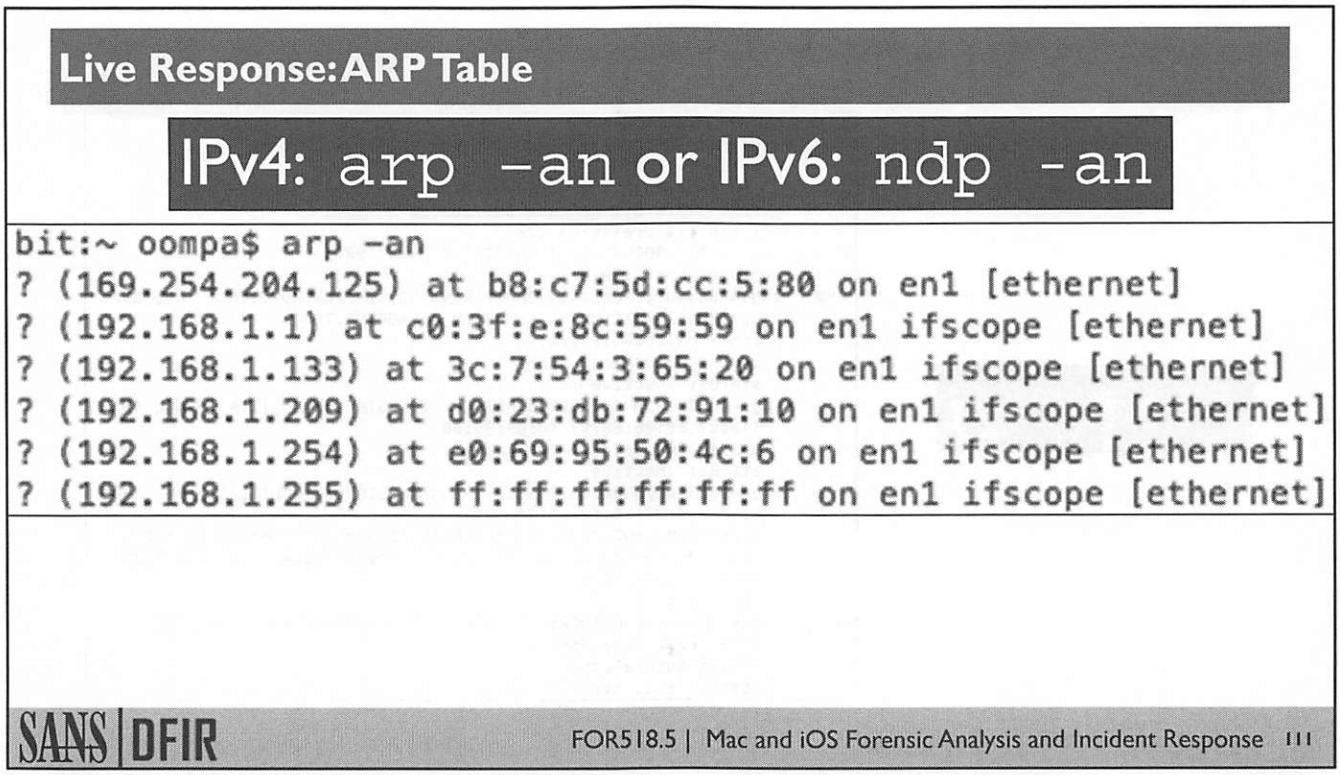

Was this system connected to a specific server recently? The `netstat -rn` command prints the routing table. The routing table can be used to determine recent network connections or static routes.

The `-r` parameter shows the routing table.

The `-n` parameter shows network addresses as numbers (rather than host and domain names).

Note: Leading 0s in MAC addresses may not show in output.

```
bit:~ oompa$ netstat -rn | more
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.254      UGSc           22        0     en1
127                127.0.0.1          UCS             0        0     lo0
127.0.0.1          127.0.0.1          UH              5    23616     lo0
169.254            link#6             UCS             1        0     en1
169.254.204.125    b8:c7:5d:cc:5:80   UHLSW           0        1     en1
192.168.1          link#6             UCS             5        0     en1
192.168.1.1        c0:3f:e:8c:59:59   UHLWIi          1      104     en1    850
192.168.1.101      127.0.0.1          UHS             0        0     lo0
192.168.1.133      3c:7:54:3:65:20    UHLWIi          0     4502     en1    295
192.168.1.209      d0:23:db:72:91:10  UHLWIi          0       45     en1   1163
192.168.1.254      e0:69:95:50:4c:6   UHLWIir        23      577     en1   1188
192.168.1.255      ff:ff:ff:ff:ff:ff  UHLWbI          0        1     en1
```



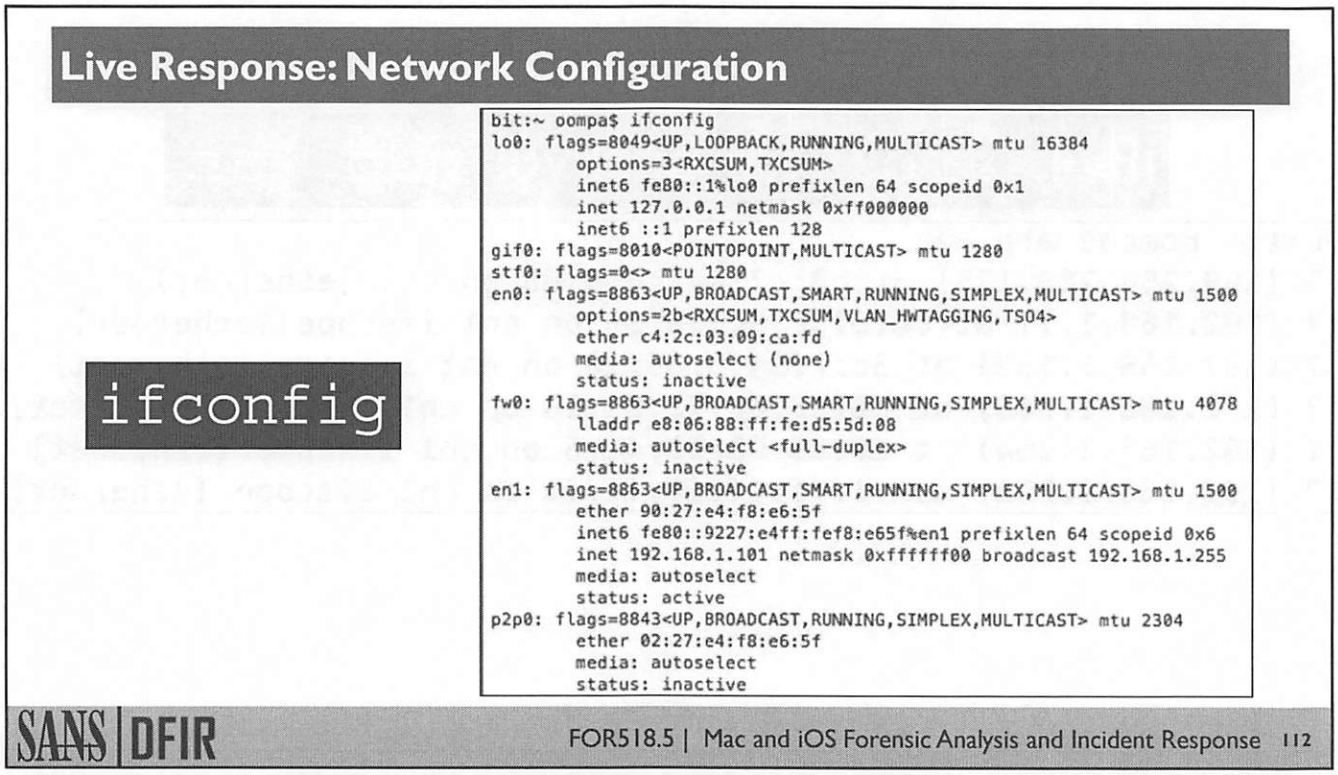
Does the target have a NAS hiding in a closet somewhere?

The `arp -an` (or `ndp -an` for IPv6) command prints the Address Resolution Protocol (ARP) table. The ARP table shows the IP address to MAC address resolution used for systems on the local network. This may aid in identifying other systems to pursue.

The `-a` parameter shows the state of all network sockets.

The `-n` parameter shows network addresses as numbers (rather than host and domain names).

Note: Leading 0s in MAC addresses may not show in output.



The `ifconfig` command prints the network configuration of the system.

In the screenshot, we can see this system has multiple network interfaces. The active interface, `en1` has an IP address of `192.168.1.101`. We can also see the various MAC addresses for each interface. This may be helpful if network logs were captured.

Network Interfaces:

- `lo#` – Loopback
- `gif#` – Generic Tunnel
- `stf#` – IPv6 to IPv4 Tunnel
- `en#` – Ethernet or Wireless: These are likely the most important to an investigator.
- `fw#` – FireWire
- `p2p#` – Point-to-Point

More information about the wireless interface can be found by using the airport command. This utility is located in a directory that is not in the default user's path; therefore, it should be run as shown below. This command can supply the station name, its MAC address, and its authentication status (i.e., WPA2, WEP).
`/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport -I`

The `ifconfig -v` command will produce an additional output starting with "`type: *`" that contains the type of connection the interface is implementing (i.e., Wi-Fi, Ethernet, etc.).

```
bit:~ oompa$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=2b<RXCSUM,TXCSUM,VLAN_HWTAGGING,TSO4>
        ether c4:2c:03:09:ca:fd
        media: autoselect (none)
        status: inactive
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
        lladdr e8:06:88:ff:fe:d5:5d:08
        media: autoselect <full-duplex>
        status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 90:27:e4:f8:e6:5f
        inet6 fe80::9227:e4ff:fef8:e65f%en1 prefixlen 64 scopeid 0x6
        inet 192.168.1.101 netmask 0xffffff00 broadcast 192.168.1.255
        media: autoselect
        status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        ether 02:27:e4:f8:e6:5f
        media: autoselect
        status: inactive
```

Live Response: Open Files

`lsof`

```
word:~ oompa$ lsof | grep Chrome
Google     8239 oompa  txt      REG               1,4      17968 16575795 /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
Google     8239 oompa  txt      REG               1,4  105742672 16575212 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4      45056 16151372 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_0
Google     8239 oompa  txt      REG               1,4     270336 16151373 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_1
Google     8239 oompa  txt      REG               1,4       8192 16151374 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_2
Google     8239 oompa  txt      REG               1,4    4202496 16151375 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_3
Google     8239 oompa  txt      REG               1,4      11642 16575321 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4      11560 16575322 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4         96 16876728 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt      REG               1,4         96 16876261 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt      REG               1,4         16 16579661 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt      REG               1,4      45056 15374902 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt      REG               1,4   10756096 15810615 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_1
Google     8239 oompa  txt      REG               1,4   13639680 15810616 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_2
Google     8239 oompa  txt      REG               1,4   62922752 15810617 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_3
Google     8239 oompa  txt      REG               1,4     262512 15810613 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/index
Google     8239 oompa  txt      REG               1,4     598016 15810614 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_0
Google     8239 oompa  txt      REG               1,4   10206320 16575335 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4     396173 16575363 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4     551484 16575384 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4     270336 15374903 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt      REG               1,4    4202496 15374905 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt      REG               1,4    1056768 15374904 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt      REG               1,4    1089968 16575295 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4    1655945 16575296 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4     250665 16575306 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4   16862438 16575375 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
Google     8239 oompa  txt      REG               1,4     826224 16575191 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Goog
```

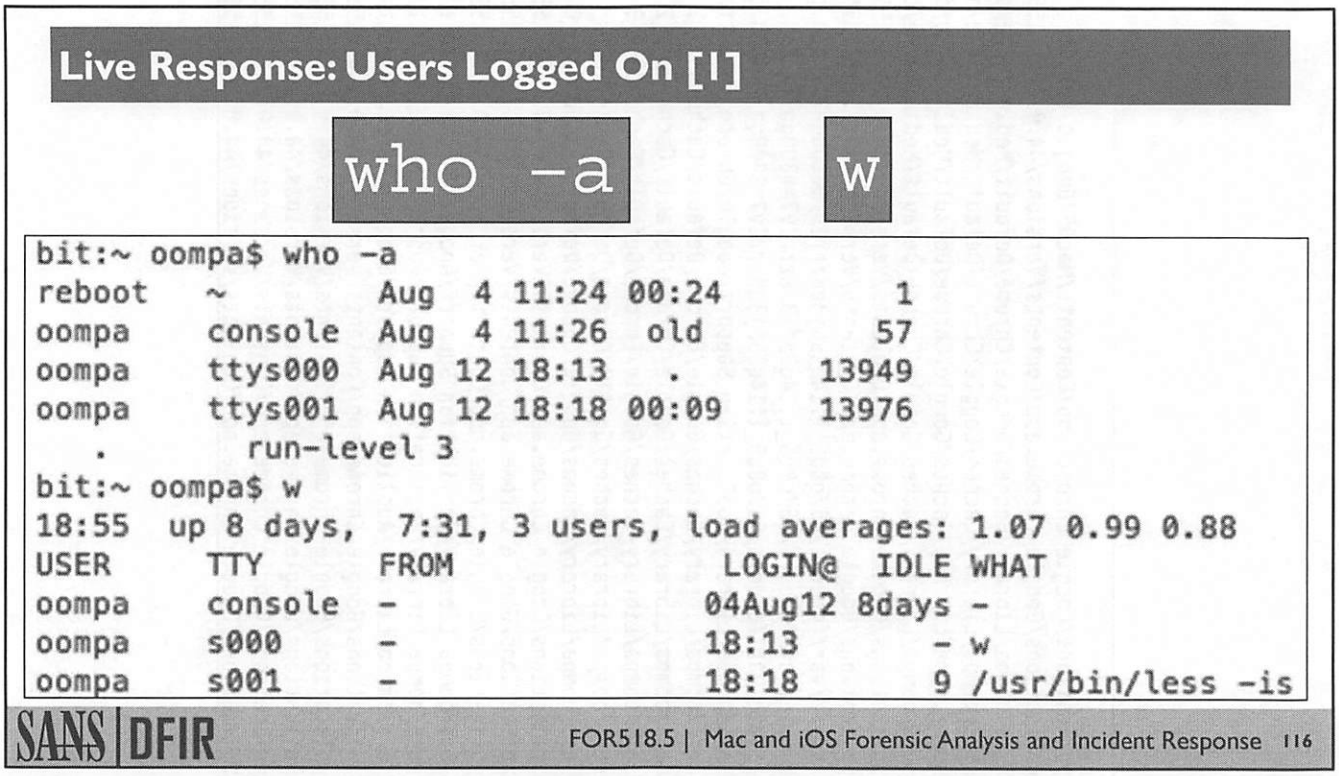

The `lsof` command prints the open files (and network connections) by the process that is using them. We can see what data files and documents the user's processes have open per application.

It includes the process ID, username, and filename.

For example, keyloggers may have an open file handle to the keylog file they are writing keystrokes to. Using the `lsof` command, we may be able to identify a rogue keylogger process, and identify the keylog file it is writing to.

In the screenshot above, the results have been filtered using the "grep" command to find items relating to the Chrome application.

```
word:~ oompa$ lsof | grep Chrome
Google     8239 oompa  txt       REG               1,4      17968 16575795 /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
Google     8239 oompa  txt       REG               1,4  105742672 16575212 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4      45056 16151372 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_0
Google     8239 oompa  txt       REG               1,4     270336 16151373 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_1
Google     8239 oompa  txt       REG               1,4       8192 16151374 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_2
Google     8239 oompa  txt       REG               1,4    4202496 16151375 /Users/oompa/Library/Caches/Google/Chrome/Default/Media Cache/data_3
Google     8239 oompa  txt       REG               1,4      11642 16575321 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4      11560 16575322 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4         96 16876728 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt       REG               1,4         96 16876261 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt       REG               1,4         16 16579661 /private/var/folders/86/q0q5_lf14q35dj3q8zrr2v7m0000gn/T/.com.google.
Google     8239 oompa  txt       REG               1,4      45056 15374902 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt       REG               1,4   10756096 15810615 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_1
Google     8239 oompa  txt       REG               1,4   13639680 15810616 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_2
Google     8239 oompa  txt       REG               1,4   62922752 15810617 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_3
Google     8239 oompa  txt       REG               1,4     262512 15810613 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/index
Google     8239 oompa  txt       REG               1,4     598016 15810614 /Users/oompa/Library/Caches/Google/Chrome/Default/Cache/data_0
Google     8239 oompa  txt       REG               1,4   10206320 16575335 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4     396173 16575363 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4     551484 16575384 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4     270336 15374903 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt       REG               1,4    4202496 15374905 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt       REG               1,4    1056768 15374904 /Users/oompa/Library/Application Support/Google/Chrome/Default/Servic
Google     8239 oompa  txt       REG               1,4    1089968 16575295 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4    1655945 16575296 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4     250665 16575306 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4   16862438 16575375 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
Google     8239 oompa  txt       REG               1,4     826224 16575191 /Applications/Google Chrome.app/Contents/Versions/44.0.2403.155/Googl
```



The `who -a` command displays the following:

- Time of last system boot
- Logged-in users
- Type of Logon
  - `console` – Login via GUI
  - `ttys*` – Login via Terminal
- User Login Time
- User Idle Time
- Associated Process ID

The `w` command has much of the same information as `who -a`; however, it also contains:

- Local System Time
- System Uptime
- Number of Users
- Remote Login System Information
- Current Activity of the Logon Process

The "`oompa`" user in the screenshot has one login process open (through the normal login screen) and two Terminal sessions open. Each Terminal session/window is a separate login process.

You may also want to identify the currently logged-in user; try these commands:

- `who am i`: Show the currently logged-in username, console number, and login process timestamp.
- `whoami`: Show the username of the currently logged-in user.
- `id`: Show the "User Identity" of the current user, including user ID (UID) and group (GID) affiliations.

Live Response: Users Logged On [2]

```
nibble:/ sledwards$ last
sledwards  ttys001                  Sat Feb  1 12:18   still logged in
sledwards  ttys000                  Fri Jan 31 20:26   still logged in
sledwards  ttys000                  Sun Jan 26 21:49 - 21:52  (00:03)
sledwards  ttys006                  Wed Jan 22 18:40 - 19:59  (01:18)
sledwards  ttys005                  Tue Jan 21 18:21 - 21:17 (7+02:55)
sledwards  ttys004                  Tue Jan 21 18:15 - 21:17 (7+03:02)
sledwards  ttys002                  Tue Jan 21 16:48 - 21:17 (7+04:29)
sledwards  ttys008                  Tue Jan 21 05:31 - 12:29  (06:57)
sledwards  ttys007                  Tue Jan 21 05:29 - 16:48  (11:18)
sledwards  ttys006                  Tue Jan 21 05:18 - 16:48  (11:29)
sledwards  ttys005                  Tue Jan 21 05:15 - 16:48  (11:33)
sledwards  ttys004                  Tue Jan 21 04:47 - 16:48  (12:01)
sledwards  ttys002                  Tue Jan 21 04:40 - 16:48  (12:07)
sledwards  ttys003                  Mon Jan 20 18:01 - 21:42 (2+03:41)
sledwards  ttys002                  Mon Jan 20 12:56 - 20:09  (07:12)
```

last

The `last` command lists the last logins of the system by user, terminal session, date, a hostname (if applicable), and session length.

The timestamp shown gives the duration of the login process. For example, "`7+02:55`" means the user has had this login process open for one week (7 days), 2 hours, and 55 minutes.

The end of the `last` command output contains the timestamp when the "`wtmp`" logging started. For example, "`wtmp begins Mon Jul 7 10:08`".

Live Response: Running Processes

`ps aux`

```
oompa          8814   0.3  1.2  1003732  98268   ??  S    Wed06PM   1:56.18 /Applications/Microsoft Office 2011/M
kelly         14583   0.2  0.5  2525132  45848   ??  S     7:09PM   0:02.11 /System/Library/PrivateFrameworks/Hel
oompa          8586   0.2  0.3  2572748  22840   ??  S    Tue08PM   4:49.34 /Applications/Utilities/Activity Moni
oompa           237   0.2  1.6  3847120 137032   ??  S     4Aug12  10:45.67 /System/Library/CoreServices/Finder.a
root           8589   0.1  0.0  2445112   3248   ??  Ss   Tue08PM   3:30.86 /usr/libexec/activitymonitord
oompa           254   0.1  0.2  2548272  20176   ??  SN    4Aug12   0:12.17 /System/Library/CoreServices/Notifica
oompa           279   0.1  0.8   758272  65752   ??  S     4Aug12  14:27.73 /Applications/Dropbox.app/Contents/Ma
charlie       14677   0.0  0.0  2484312   3532   ??  SN    7:10PM   0:00.12 /usr/sbin/usernoted
charlie       14676   0.0  0.0  2485232   3000   ??  S     7:10PM   0:00.07 /System/Library/CoreServices/NetworkB
charlie       14673   0.0  0.2  2531632  16512   ??  S     7:10PM   0:00.48 /System/Library/CoreServices/Finder.a
charlie       14672   0.0  0.6  2580296  51196   ??  S     7:10PM   0:04.97 /System/Library/CoreServices/Dock.app
charlie       14662   0.0  0.0  2488108   2472   ??  S     7:09PM   0:00.03 /System/Library/Services/AppleSpell.s
charlie       14652   0.0  0.0  2467404   2920   ??  S     7:09PM   0:00.07 /System/Library/CoreServices/pbs
charlie       14651   0.0  0.3  2599168  24472   ??  S     7:09PM   0:01.43 /System/Library/CoreServices/SystemUI
charlie       14649   0.0  0.1  2501732  10120   ??  S     7:09PM   0:01.18 /System/Library/Frameworks/Applicatio
charlie       14648   0.0  0.1  2504224   7332   ??  S     7:09PM   0:00.11 /System/Library/CoreServices/talagent
charlie       14642   0.0  0.0  2433976   1592   ??  S     7:09PM   0:00.08 /usr/sbin/pboard
charlie       14633   0.0  0.0  2483820   2456   ??  S     7:09PM   0:00.13 /usr/sbin/distnoted agent
charlie       14631   0.0  0.0  2465740   1500   ??  S     7:09PM   0:00.25 /usr/sbin/cfprefsd agent
charlie       14629   0.0  0.0  2471500   1576   ??  Ss    7:09PM   0:00.14 /sbin/launchd
charlie       14616   0.0  0.3  2572380  25728   ??  Ss    7:09PM   0:01.50 /System/Library/CoreServices/loginwi
kelly         14608   0.0  0.0  2465392   2036   ??  S     7:09PM   0:00.02 /System/Library/CoreServices/AirPort
kelly         14603   0.0  0.2  2516072  14220   ??  Ss    7:09PM   0:00.31 com.apple.dock.extra
```


The `ps aux` command outputs the process status. The output contains information such as (see the man page for more detail):

- User Account
- Process ID
- Percentage of CPU and RAM
- Process Start Date and Time
- Issued Command

Note the lack of the '-' in the command. Unlike other systems, putting this dash in the command will be unsuccessful. This is a good example of how OS X commands may be different than other UNIX-like systems.

You may want to experiment with other command arguments, such as the following:

- `-ww`: Use this flag to display all the information; otherwise, it defaults to screen width.
- `-ef`: Display more information, including parent process IDs.

```
oompa      8814  0.3  1.2  1003732  98268  ??  S     Wed06PM   1:56.18 /Applications/Microsoft Office 2011/M
kelly     14583  0.2  0.5  2525132  45848  ??  S      7:09PM   0:02.11 /System/Library/PrivateFrameworks/He
oompa      8586  0.2  0.3  2572748  22840  ??  S     Tue08PM   4:49.34 /Applications/Utilities/Activity Mon
oompa       237  0.2  1.6  3847120 137032  ??  S      4Aug12  10:45.67 /System/Library/CoreServices/Finder.a
root       8589  0.1  0.0  2445112   3248  ??  Ss    Tue08PM   3:30.86 /usr/libexec/activitymonitord
oompa       254  0.1  0.2  2548272  20176  ??  SN     4Aug12   0:12.17 /System/Library/CoreServices/Notifica
oompa       279  0.1  0.8   758272  65752  ??  S      4Aug12  14:27.73 /Applications/Dropbox.app/Contents/Ma
charlie   14677  0.0  0.0  2484312   3532  ??  SN     7:10PM   0:00.12 /usr/sbin/usernoted
charlie   14676  0.0  0.0  2485232   3000  ??  S      7:10PM   0:00.07 /System/Library/CoreServices/NetworkB
charlie   14673  0.0  0.2  2531632  16512  ??  S      7:10PM   0:00.48 /System/Library/CoreServices/Finder.a
charlie   14672  0.0  0.6  2580296  51196  ??  S      7:10PM   0:04.97 /System/Library/CoreServices/Dock.app
charlie   14662  0.0  0.0  2488108   2472  ??  S      7:09PM   0:00.03 /System/Library/Services/AppleSpell.s
charlie   14652  0.0  0.0  2467404   2920  ??  S      7:09PM   0:00.07 /System/Library/CoreServices/pbs
charlie   14651  0.0  0.3  2599168  24472  ??  S      7:09PM   0:01.43 /System/Library/CoreServices/SystemUI
charlie   14649  0.0  0.1  2501732  10120  ??  S      7:09PM   0:01.18 /System/Library/Frameworks/Applicatio
charlie   14648  0.0  0.1  2504224   7332  ??  S      7:09PM   0:00.11 /System/Library/CoreServices/talagent
charlie   14642  0.0  0.0  2433976   1592  ??  S      7:09PM   0:00.08 /usr/sbin/pboard
charlie   14633  0.0  0.0  2483820   2456  ??  S      7:09PM   0:00.13 /usr/sbin/distnoted agent
charlie   14631  0.0  0.0  2465740   1500  ??  S      7:09PM   0:00.25 /usr/sbin/cfprefsd agent
charlie   14629  0.0  0.0  2471500   1576  ??  Ss     7:09PM   0:00.14 /sbin/launchd
charlie   14616  0.0  0.3  2572380  25728  ??  Ss     7:09PM   0:01.50 /System/Library/CoreServices/loginwin
kelly     14608  0.0  0.0  2465392   2036  ??  S      7:09PM   0:00.02 /System/Library/CoreServices/AirPort
kelly     14603  0.0  0.2  2516072  14220  ??  Ss     7:09PM   0:00.31 com.apple.dock.extra
```

Live Response: System Profiler

```
system_profiler -xml -detaillevel full >
/Volumes/IR_CASE/sys_prof_MBP.spx
```

Open in "System Information.app" or system_profiler command

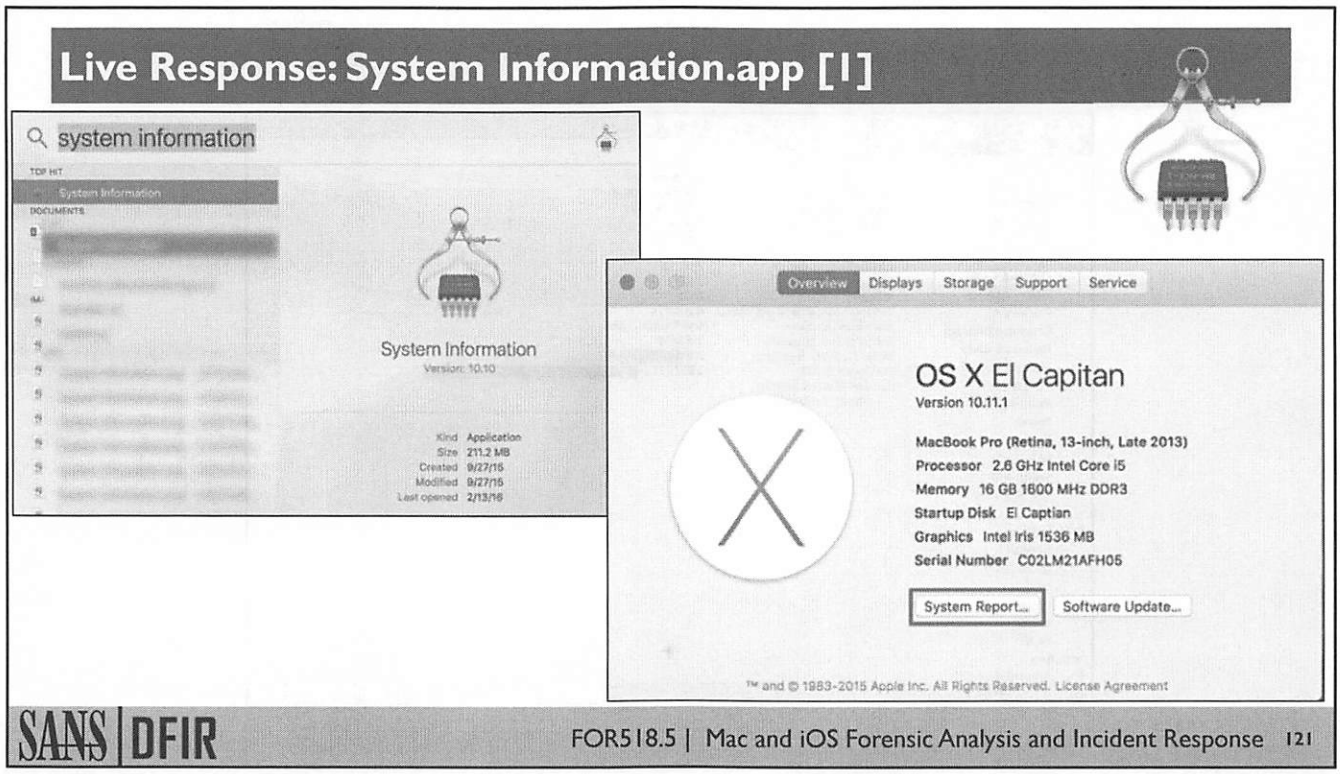
- Hardware Information
- USB Information
- Network Information
- Firewall Settings
- Mounted Volumes
- System Information
- Applications
- Kernel Extensions
- Log Data

The `system_profiler` command is the command-line version of the System Information application.

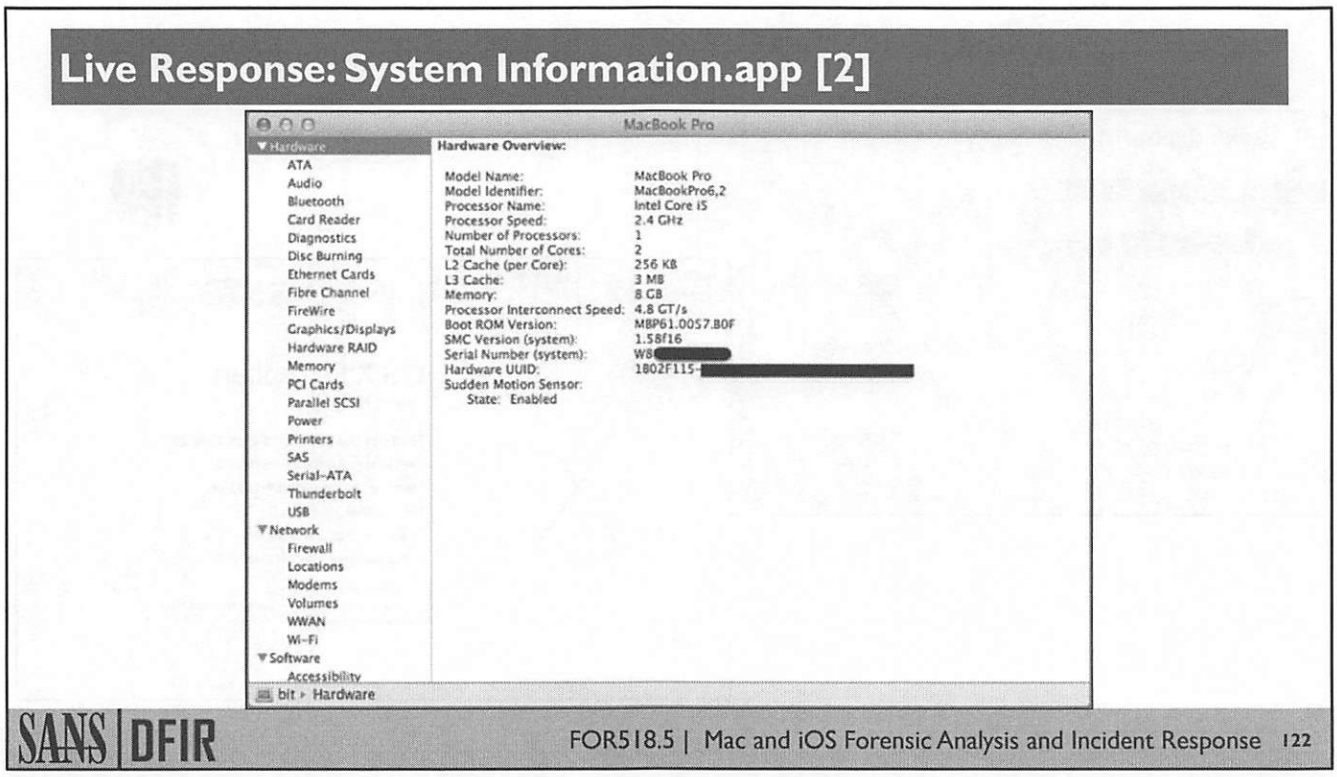
The `system_profiler` command has different output formats and detailed level options. As long as the output is named with a ".`spx`" file extension, the data can be viewed in the System Information.app. The XML output option can be used with the `System Information.app` located in `/Applications/Utilities/`. The output levels can be set as "`mini`", "`basic`", and "`full`", or by the specific data type that can be accessed by the command `system_profiler -listDataTypes`, which will produce the following output.

Available Datatypes:

```
SPParallelATADataType
SPUniversalAccessDataType
SPApplicationsDataType
SPAudioDataType
SPBluetoothDataType
SPCardReaderDataType
SPComponentDataType
SPDeveloperToolsDataType
SPDiagnosticsDataType
SPDiscBurningDataType
SPEthernetDataType
SPExtensionsDataType
SPFibreChannelDataType
SPFireWireDataType
SPFirewallDataType
SPFontsDataType
SPFrameworksDataType
SPDisplaysDataType
SPHardwareDataType
...
```



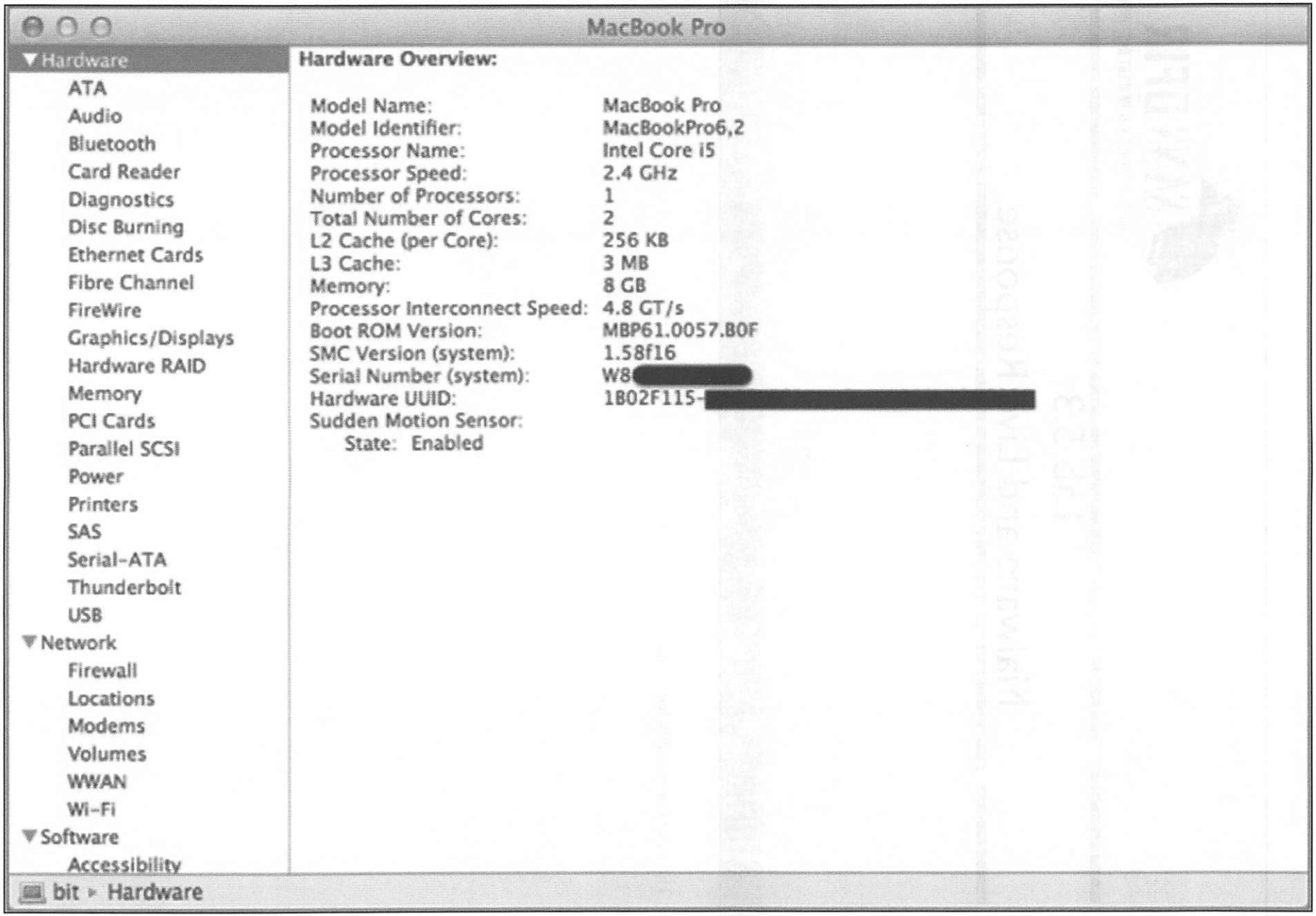
These screenshots show how to access the System Information.app application from the Spotlight menu or from the "About This Mac" menu, available by selecting the Apple logo in the top left corner of the screen and selecting "System Report".

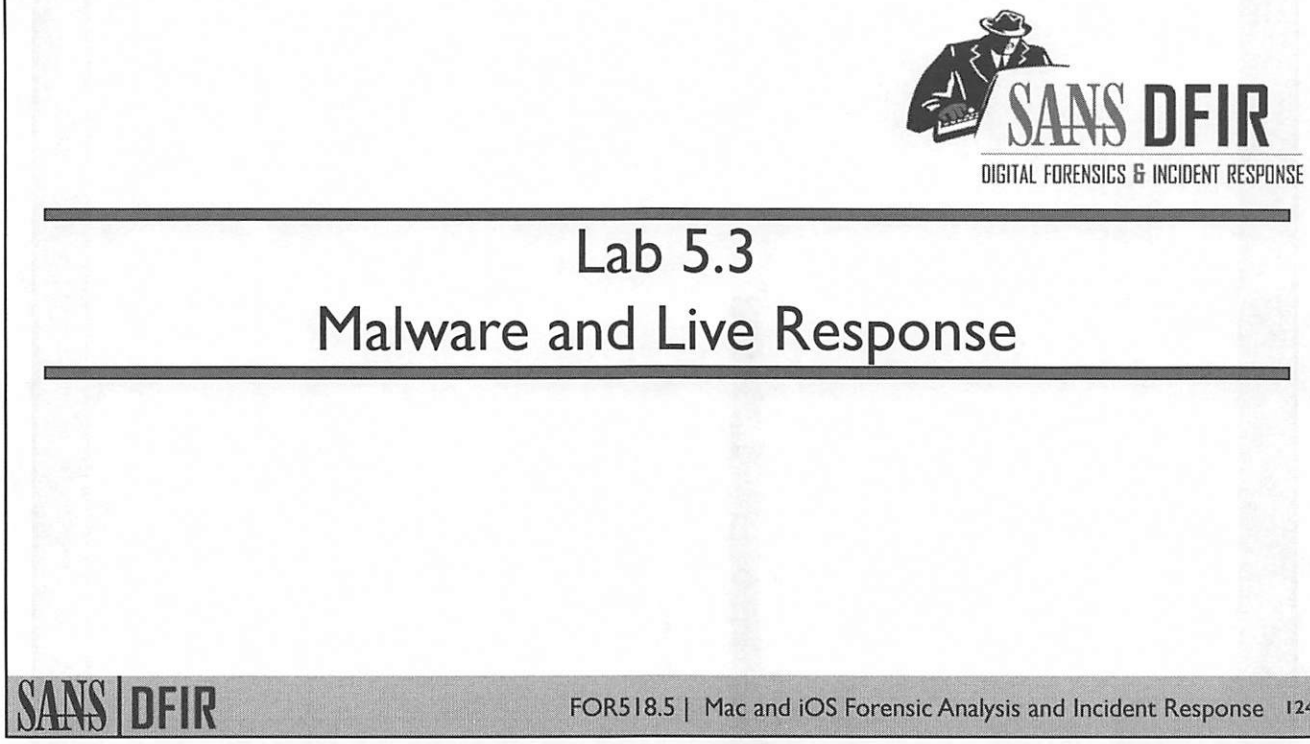


This screenshot shows the types of data available from the `System Information.app` application or from the `system_profiler` command.

This tool is native to OS X.

You may also open the `System Information.app` application and choose `File | Save` to save this information directly from the GUI rather than on the command line.



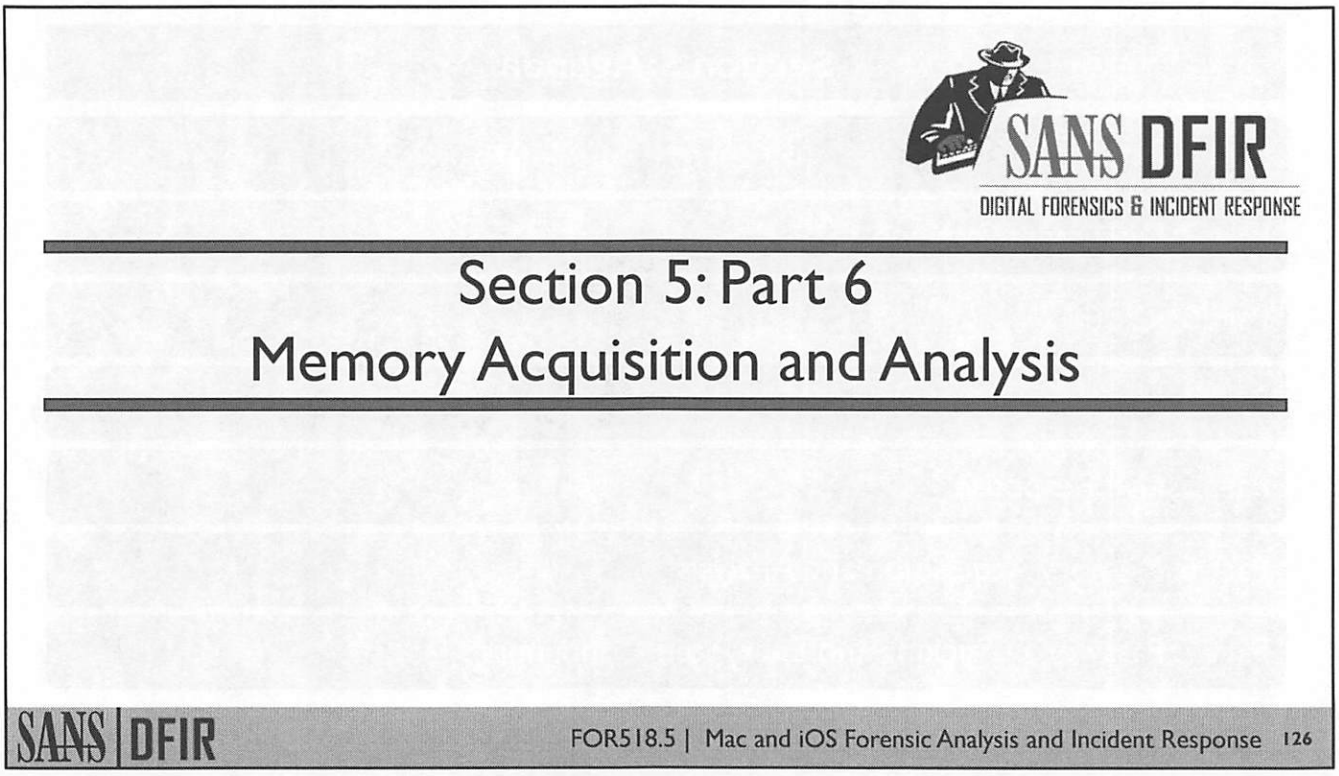


This page intentionally left blank.

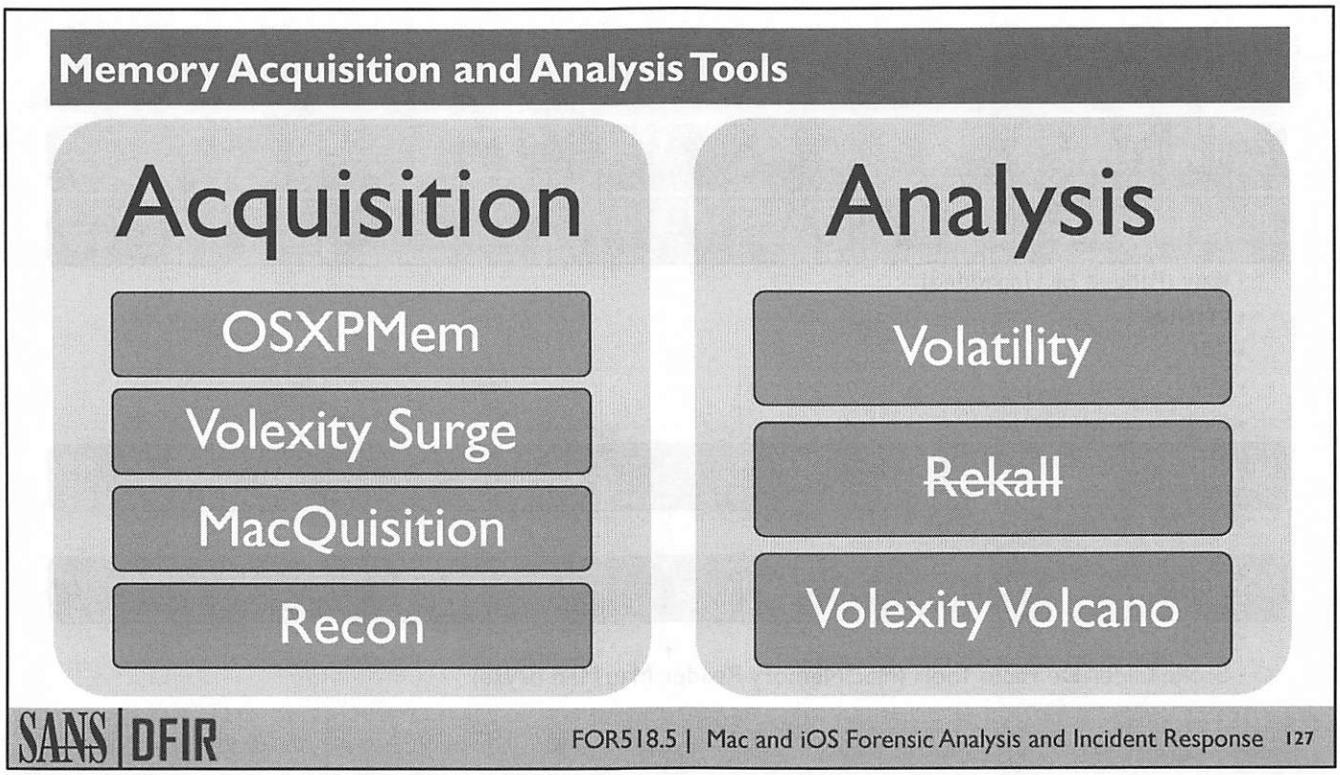
Section 5: Agenda

- Part 1: Pattern of Life
- Part 2: Document Versions
- Part 3: iCloud
- Part 4: Malware and Intrusion Analysis
- Part 5: Live Response
- Part 6: Memory Acquisition and Analysis
- Part 7: Password Cracking and Encrypted Containers

This page intentionally left blank.



This page intentionally left blank.

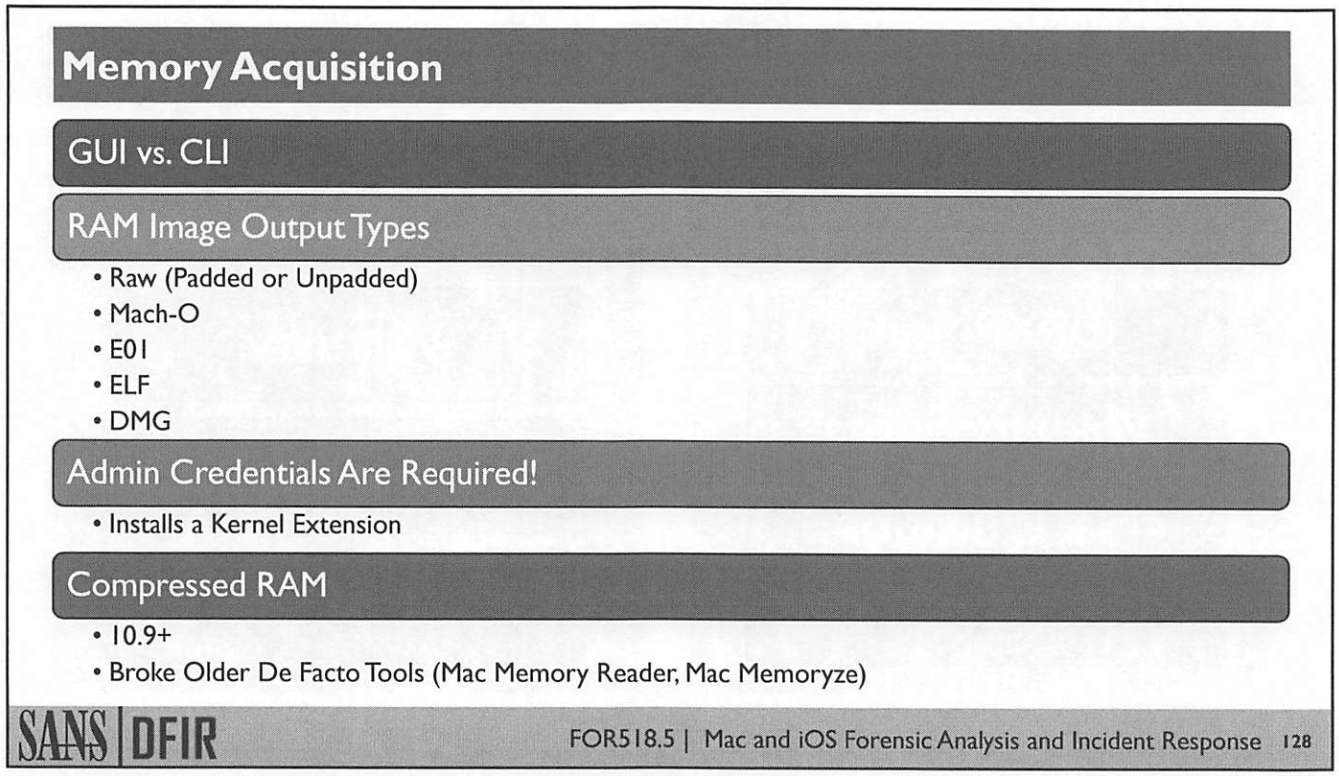


Capturing and analyzing memory from Windows-based systems has been available for years, while Mac memory acquisition and analysis is relatively new.

In the past couple of years or so, a few developers have been able to provide us the capability to acquire RAM from Mac systems as well as the ability to analyze these RAM images. A mix of free, open-source, and commercial tools are available to perform these tasks.

Each tool has their advantages and disadvantages:

- Some are free.
- Some are easier to use.
- Some have more capabilities.
- Some are updated by the developers more frequently.



Memory acquisition is the activity of capturing the system random access memory (RAM). This RAM capture contains process lists, network information, open files, and other data that may be useful in forensic analysis.

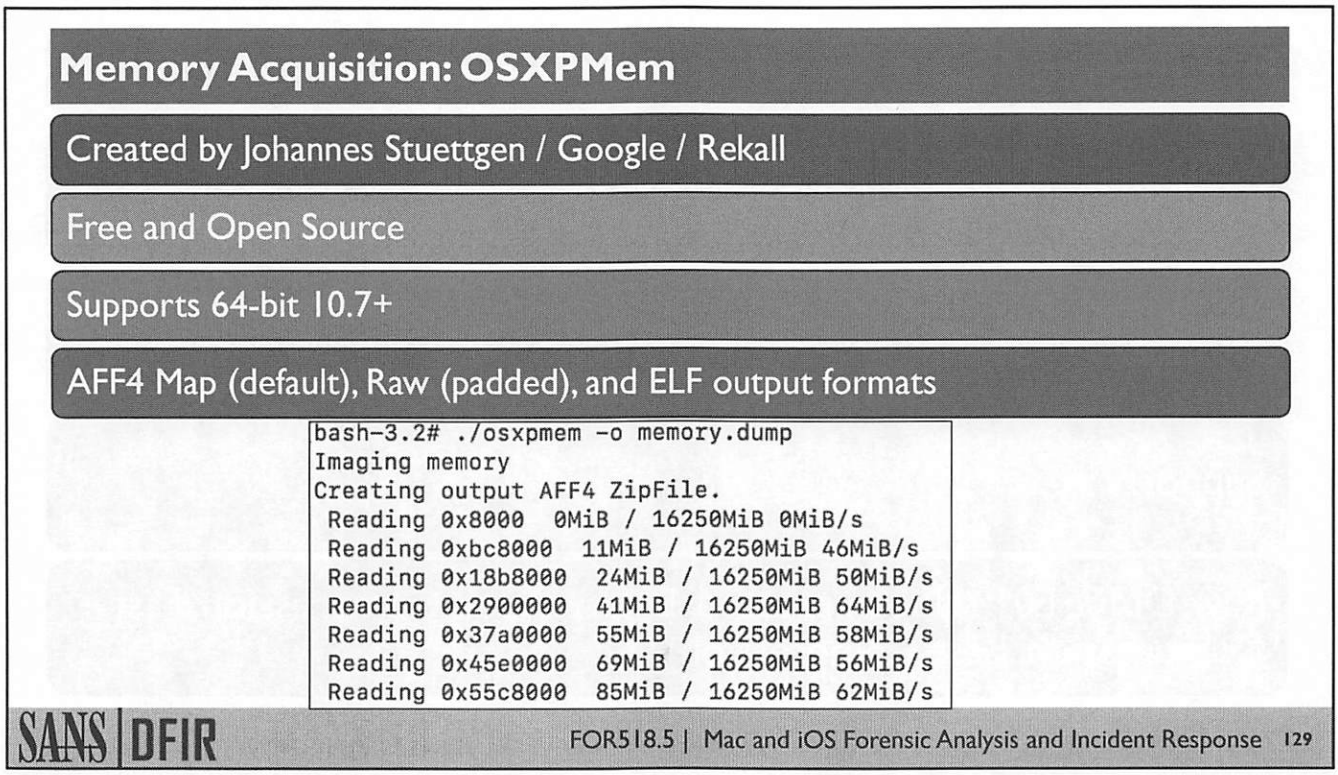
These tools have either GUI applications or command-line tools. It is worth noting that a GUI application may leave a larger memory footprint than a command-line tool.

Each tool captures the RAM in their own specified formats; some tools have multiple output options, while others are one and done. An analyst must be aware of the output and their preferred analysis tool. One output type may not be supported in another analysis tool. Administrator credentials are required to dump RAM, as all tools install a kernel extension.

Common output formats include:

- Raw (padded and unpadded): DD-style format—padded format adds zeros where there are unmapped memory regions. Final RAM image size may be larger than RAM in the system. Raw/padded is one of the de facto image standards.
- Mach-O: Mach-O file of RAM, standard format. This format is another one of the de facto image standards.
- E01: Format used with Encase forensic software
- ELF: Executable and Linking Format
- DMG: Disk Image format, similar to Raw and padded format

10.9 introduced compressed RAM. This broke many of the de facto memory acquisition tools, such as Mac Memory Reader and Mac Memoryze. If you want to learn more about how this works, read "In Lieu of Swap: Analyzing Compressed RAM in OS X and Linux".
`https://www.dfrws.org/conferences/dfrws-usa-2014/sessions/lieu-swap-analyzing-compressed-ram-mac-os-x-and-linux`



OSXPMem is a free and open-source tool created by Johannes Stuettgen. This tool is reported to support 64-bit systems from 10.7 to 10.10.

To use OSXPMem, follow these instructions:

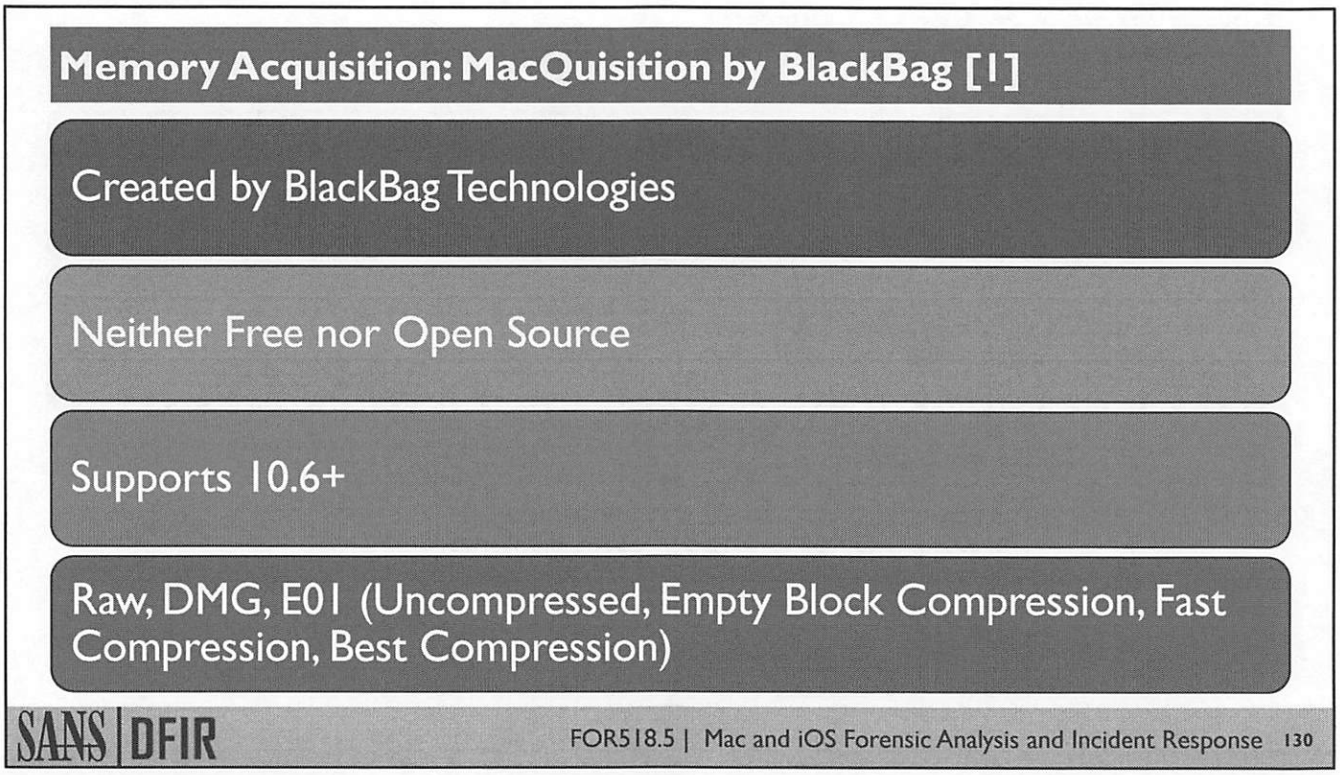
- Download the `OSXPMem-*.Zip`
- Unpack it with root privileges (use `sudo -s` or `sudo su` to get a root shell) or `chown` the `*.kext` directory to `root:wheel permissions`.
  - Still in the root shell, execute the file `osxpmem`—default output is an ELF file. Use the `-f` flag with `mach` or `raw` to get different output formats (if no longer in the root shell, use the sudo command).

```
./osxpmem -o memory.dump
```

References:

`http://releases.rekall-forensic.com/`

`http://rekall-forensic.blogspot.com/2014/03/osx-109-memory-acquisition.html`

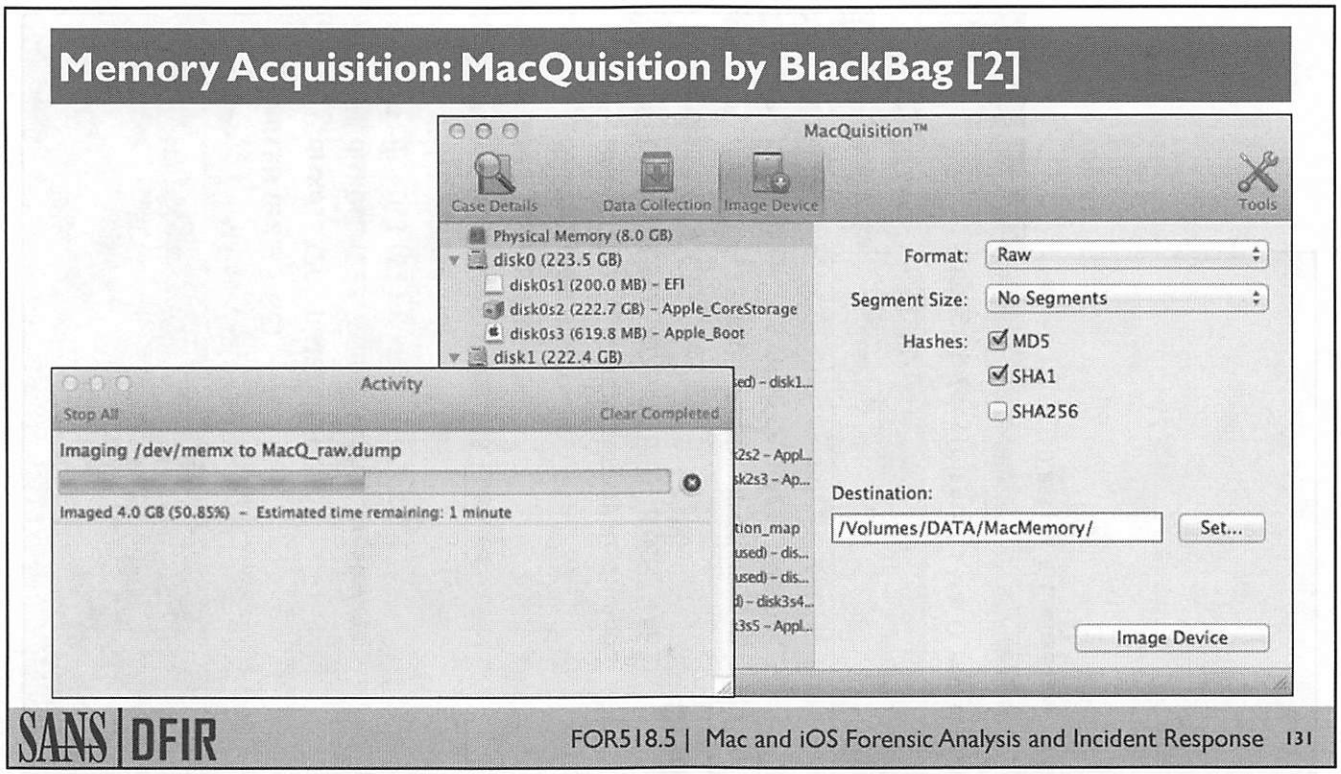


MacQuisition by BlackBag Technologies is their tool to capture all types of data in one shot. This tool allows an investigator to acquire incident response information, disk images, and RAM. While this tool makes it incredibly easy to capture everything you may need for an investigation, it does come at a price. This tool is available from `blackbagtech.com`.

This tool is only available in a GUI format that is run from a dongle-based thumb drive. RAM output formats consist of Raw/Padded, DMG, and various E01 formats.

Reference:

`https://www.blackbagtech.com/software-products/macquisition.html`

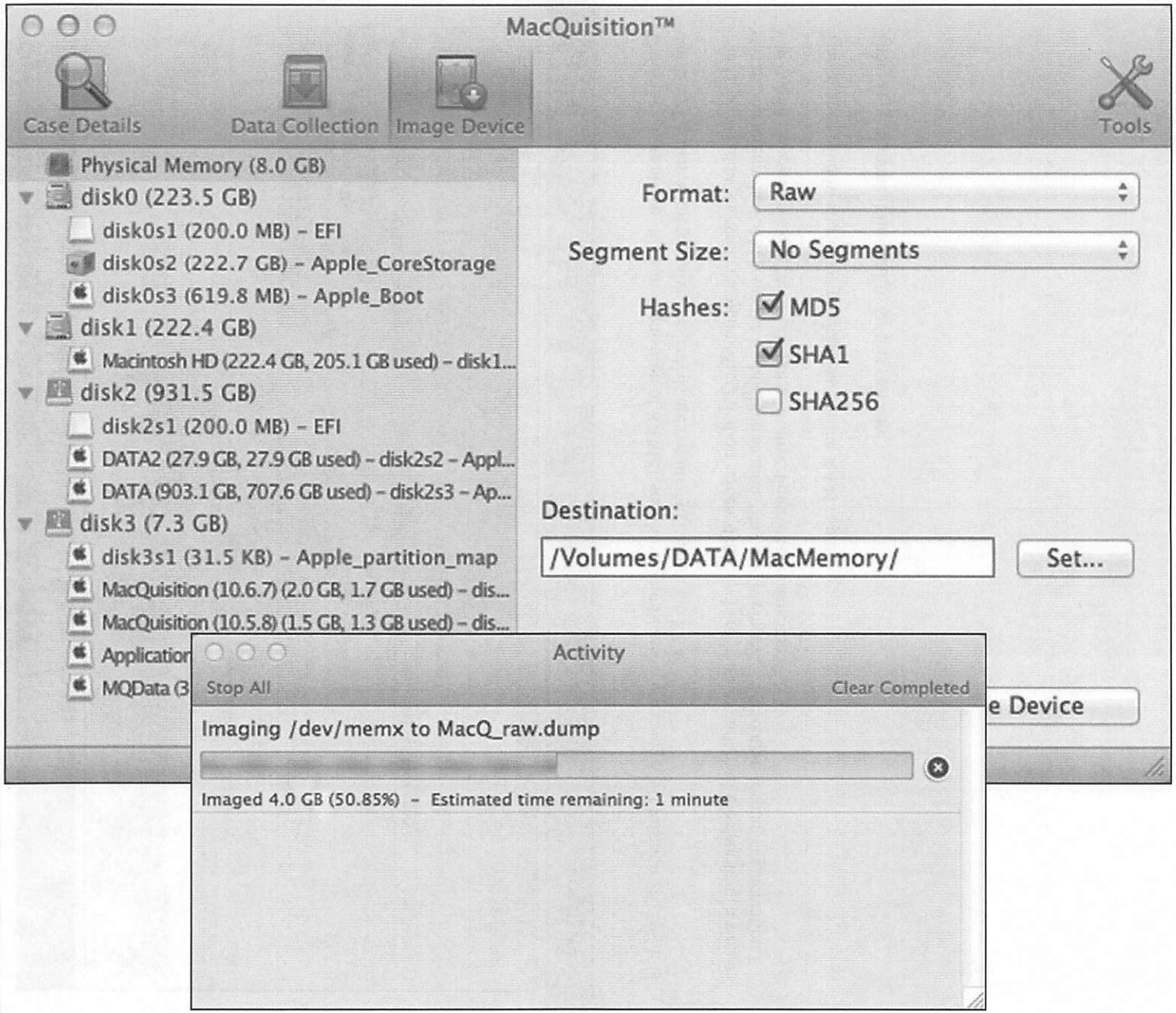


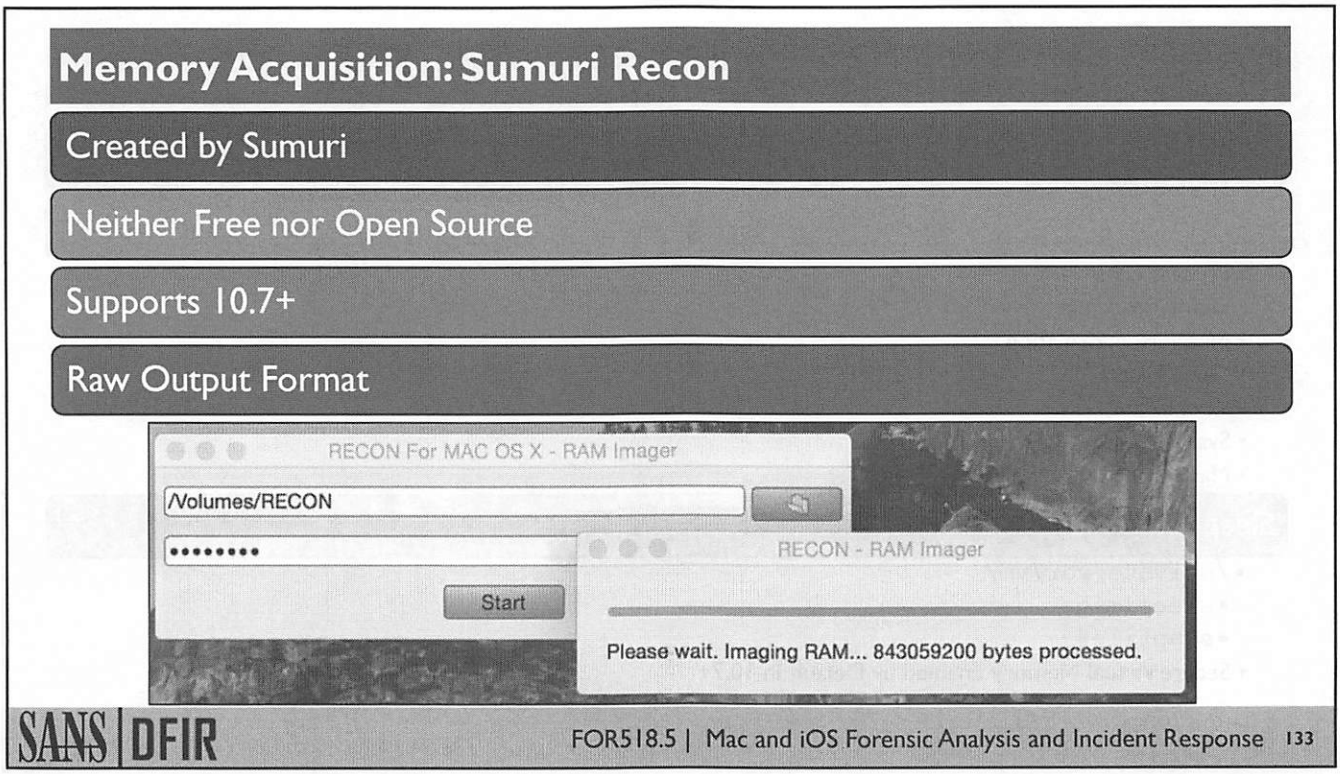
The screenshots above show an acquisition of RAM from a Mountain Lion system using MacQuisition 2012r3. It is worth noting that Administrative credentials are required to capture the RAM.

The larger window shown is the main MacQuisition screen. This is where you can gather all the data from a single system, including the RAM (note the highlighted section "Physical Memory"). The output format, segmentation, and output directory options can be altered to fit investigative requirements.

The smaller window shows the progress bar for the RAM capture.

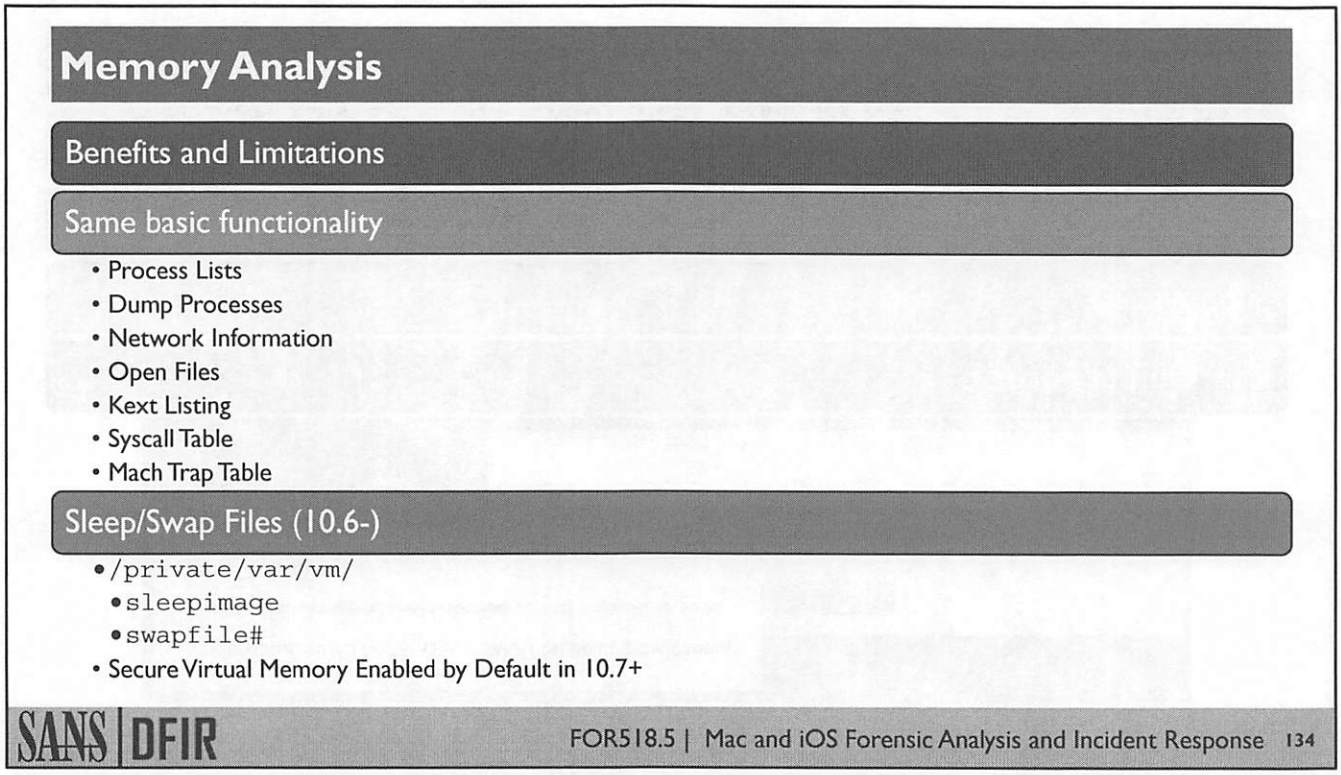
There is no command-line utility for this software.





Sumuri Recon is another all-encompassing suite of tools that can acquire disk images, RAM, triage analysis, and live response data. It does all of this in a GUI with no command-line interface. All this functionality does come at a price and is available at `sumuri.com`.

References:
`http://sumuri.com/category/recon`
`http://sumuri.com/using-recon-to-image-ram/`
`http://sumuri.com/recon-adds-support-for-mac-os-x-10-10/`



Memory analysis is the act of analyzing the memory to parse out data structures to determine what was happening on the system at that particular moment in time.

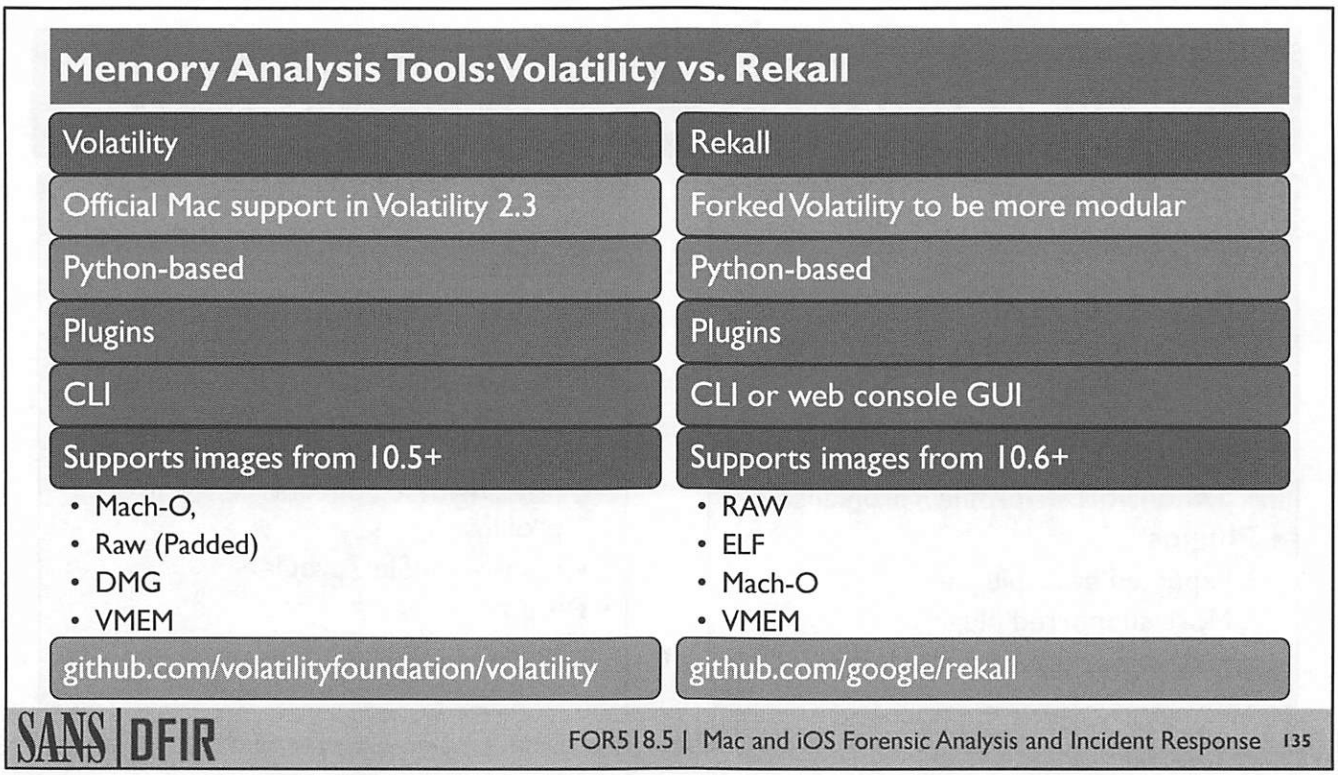
Each tool has its benefits and limitations—some tools are easier to use and some have more advanced functionality. Each tool has the same basic functionality that includes parsing the data structure to access the following information.

- Process Lists
- Dump Processes
- Network Information
- Open Files
- Kext Listing
- Syscall Table
- Mach Trap Table

Warning Caveat: Output for each image/tool combination type may vary. It doesn't hurt to perform analysis using multiple tools!

Similar to memory are hibernation and swap files. In their unencrypted formats, these files may contain passwords and other sensitive data.

The `sleepimage` file is similar to the `hiberfil.sys` file on Windows. It is created when the system goes into hibernation mode. The swap files may also contain sensitive information in their non-encrypted form. Swap files are used similarly to the Windows `pagefile.sys`. Unfortunately, since 10.7, these files are encrypted by default, and passwords (or other sensitive data) strings are no longer available. If Secure Virtual Memory is not enabled, the user may use the `strings` command to find passwords and other non-encrypted strings that may be of use in an investigation.



The de facto standard in memory analysis, Volatility is a Python-based tool that can be used on any OS that supports Python.

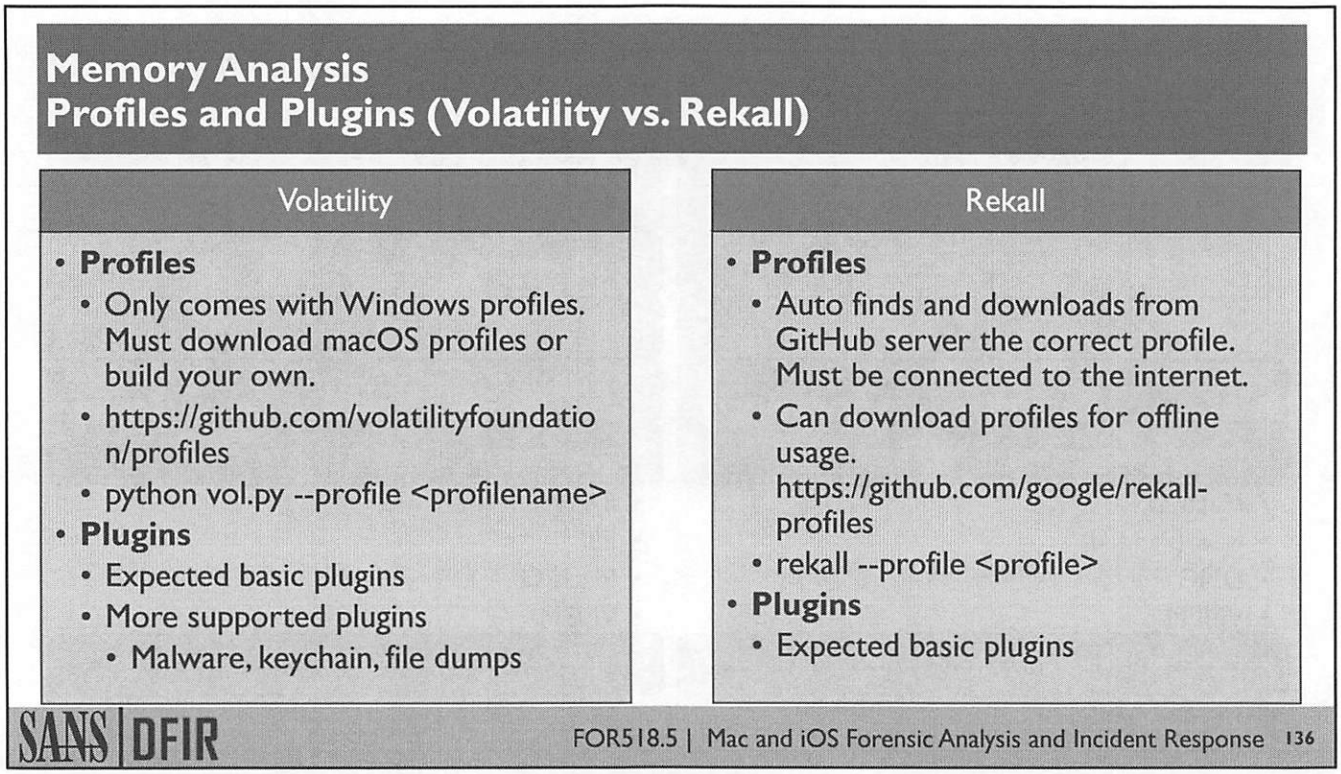
While this tool also supports images from Windows, Linux, and Android systems, it started to support Mac systems in version 2.3. Images from 10.5–10.9.x are supported with the image output formats listed above.

To make note of this again, download from the source for the latest updates that have not yet been compiled in downloaded binaries or packages.

Rekall is a fork from Volatility so that it could be built with more modularity. As with Volatility, it is Python-based and has various plugins run to gather specific data types.

Rekall is based as a command-line utility but can also be run from a web-based console GUI.

References:
`https://github.com/volatilityfoundation/volatility`
`http://www.rekall-forensic.com/`
`https://github.com/google/rekall`



Profiles are what determine where certain data structures are located and how to look for them. For OS X, a profile is required for each sub-version (i.e., the profile for 10.9.4 is different than 10.9.3). Volatility and Rekall differ in how they get the profile for the memory image you are attempting to look at.

Volatility does not come with OS X profiles by default; you will need to download the profiles from `https://github.com/volatilityfoundation/profiles` and place them in your `/volatility/plugins/overlays/mac/` directory. To ensure you have them installed correctly, run the command `python vol.py --info | grep Mac`. If no profile exists yet, you can build your own using the instructions provided here:
`https://github.com/volatilityfoundation/volatility/wiki/Mac`.

Rekall on the other hand, will automatically determine and pull the correct profile from their repository here: `https://github.com/google/rekall-profiles`. You can clone their repository and use on an offline system if required.

The basic expected plugins exist for both Volatility and Rekall, such as process listing, kext lists, open files, and network configuration. Volatility has done a better job with specialty plugins such as malware detection, keychain key dumps, and file dumping.

References:
`https://github.com/volatilityfoundation/volatility/wiki/Mac`
`https://github.com/volatilityfoundation/profiles`
`https://github.com/google/rekall-profiles/`
`http://www.rekall-forensic.com/`

Memory Analysis
Volatility Profile Installation and Check

Copy macOS specific profiles to `/volatility/plugins/overlays/mac/` directory.

- Download from: https://github.com/volatilityfoundation/profiles/tree/master/Mac

Show installed profiles

- `python vol.py --info | grep Mac`

```
word:volatility oompa$ python vol.py --info | grep Mac
Volatility Foundation Volatility Framework 2.4
MacMavericks_10_9_1_AMDx64          - A Profile for Mac Mavericks_10.9.1_AMD x64
MacMavericks_10_9_2_13C1021_AMDx64 - A Profile for Mac Mavericks_10.9.2_13C1021.AMD x64
MacMavericks_10_9_2__13C64_AMDx64  - A Profile for Mac Mavericks_10.9.2__13C64.AMD x64
MacMavericks_10_9_3_AMDx64          - A Profile for Mac Mavericks_10.9.3_AMD x64
MacMavericks_10_9_4_AMDx64          - A Profile for Mac Mavericks_10.9.4_AMD x64
MacMavericks_10_9_5_AMDx64          - A Profile for Mac Mavericks_10.9.5_AMD x64
MachOAddressSpace             - Address space for mach-o files to support atc-ny memory reader
mac_version                 - Prints the Mac version
machoinfo                   - Dump Mach-O file format information
```

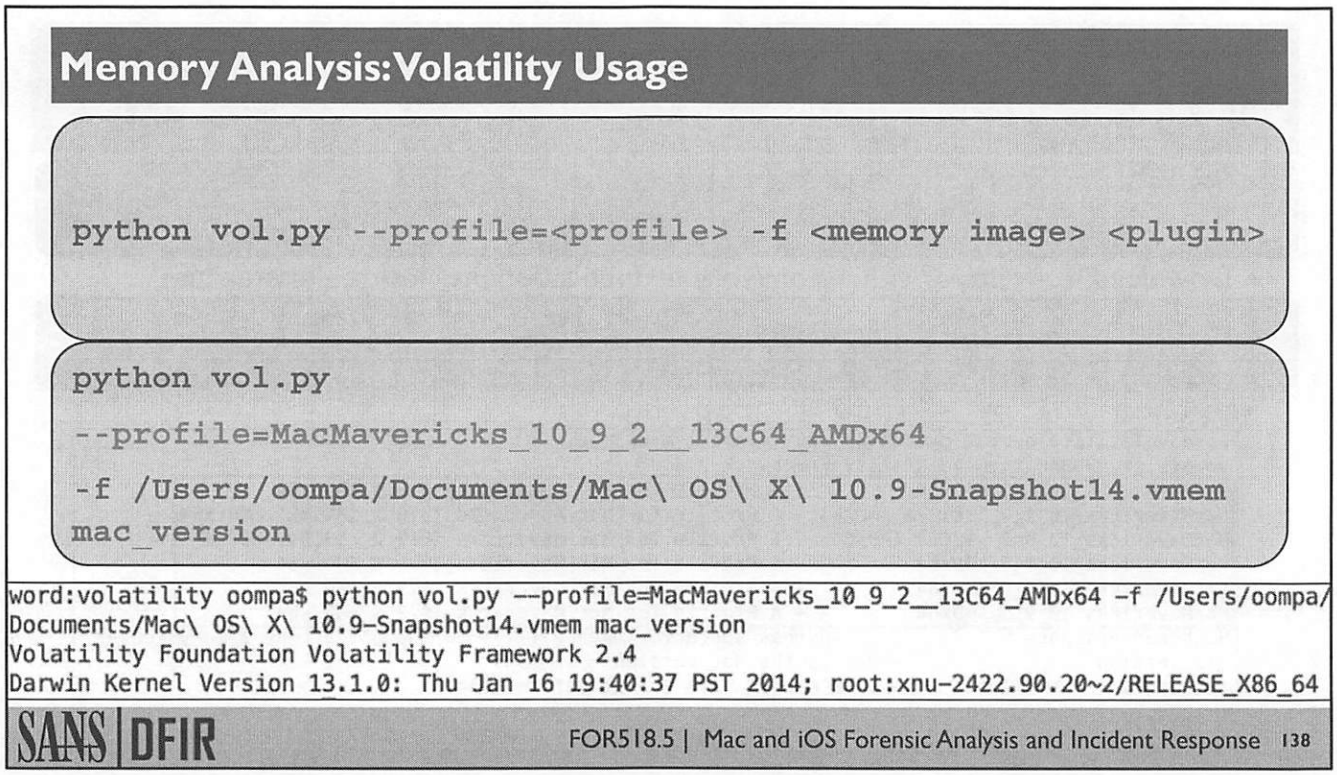

Volatility uses profiles to determine where certain data structures are to present them to the user. These profiles are downloaded and installed in the `/volatility/plugins/overlays/mac/` directory in your `Volatility` directory.

Once the files are copied into this directory, you can run the following command to determine if they are found by Volatility.

```
python vol.py --info | grep Mac
```

As shown in the screenshot, you should be able to see the names of the profiles that you have just copied into the `/volatility/plugins/overlays/mac/` directory.

Reference:
`https://github.com/volatilityfoundation/volatility/wiki/Mac`



Inputs to Volatility include, at a minimum, a file path to a memory image and a plugin name to run. For Mac memory analysis, we'll need to specify a profile, as described in the previous slide.

The example uses a profile for a specific build of 10.9.2 (Mavericks). The memory image is located in the user's `Documents` directory. This image is from a VMware Fusion virtual machine snapshot (VMEM). The analyst is attempting to determine the basic system information for this memory image. This output is similar to the `uname -a` command.

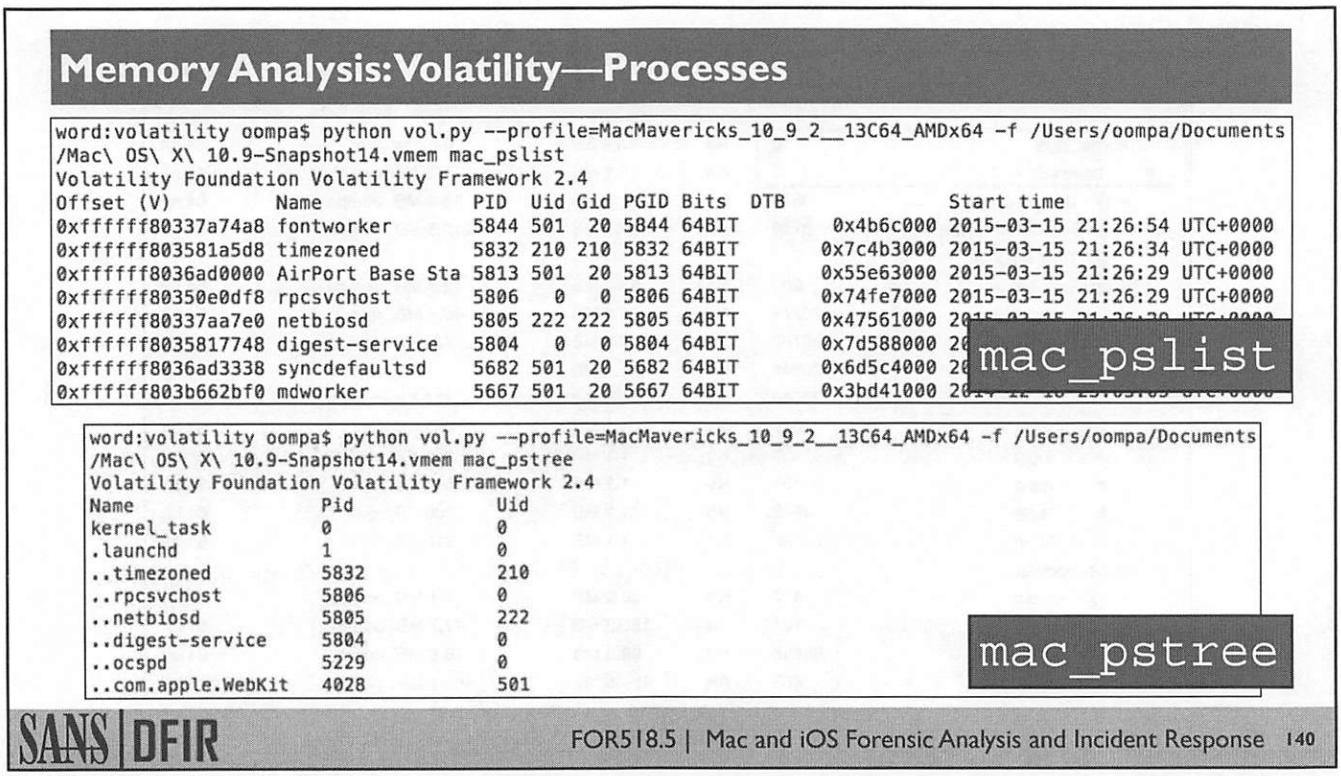
```
python vol.py --profile=MacMavericks_10_9_2__13C64_AMDx64 -f
/Users/oompa/Documents/Mac\ OS\ X\ 10.9-Snapshot14.vmem mac_pslist
```

Memory Analysis: macOS Processes

Process Name	PID	Sandbox	Memory	Compressed Mem	User	Kind
kernel_task	0	No	1.24 GB	0 bytes	root	64 bit
launchd	1	No	15.1 MB	148 KB	root	64 bit
Google Chrome	3572	No	225.4 MB	15.3 MB	oompa	64 bit
Microsoft PowerPoint	3759	No	474.5 MB	212.5 MB	oompa	32 bit
Time Machine		-	-	-	-	-
Terminal	201	No	93.4 MB	12.3 MB	oompa	64 bit
xmount	20774	No	51.6 MB	39.8 MB	root	64 bit
diskimages-helper	20782	No	7.8 MB	728 KB	oompa	64 bit
login	20648	No	1.1 MB	1.0 MB	root	64 bit
bash	20649	No	668 KB	484 KB	oompa	64 bit
login	4019	No	1.1 MB	696 KB	root	64 bit
login	41437	No	1.1 MB	924 KB	root	64 bit
login	20751	No	1.1 MB	848 KB	root	64 bit
login	3978	No	1.1 MB	900 KB	root	64 bit
login	55720	No	1.1 MB	916 KB	root	64 bit
Spotlight		-	-	-	-	-
Dropbox	417	No	86.9 MB	1.1 MB	oompa	32 bit
Mail	195	Yes	180.2 MB	17.7 MB	oompa	64 bit
LittleSnapper	55868	No	98.8 MB	18.6 MB	oompa	64 bit
TextEdit	203	Yes	184.8 MB	163.7 MB	oompa	64 bit

On a default system, all processes will be a subprocess of both `kernel_task` (always PID 0) and `launchd` (always PID 1). These parent processes are shown above in a screenshot taken from `Activity Monitor.app`.

Child processes for each application, agent, or daemon will be listed under `launchd`. In the screenshot example, the `Terminal` application has multiple `login` processes. Each of these `login` processes, in turn, has a child `bash` process.



Volatility has many different plugins. Two plugins for viewing system processes are `mac_pslist` and `mac_pstree`.

The top screenshot shows the `mac_pslist` plugin. This plugin prints the processes that were in use at the time the image was captured. Output information includes:

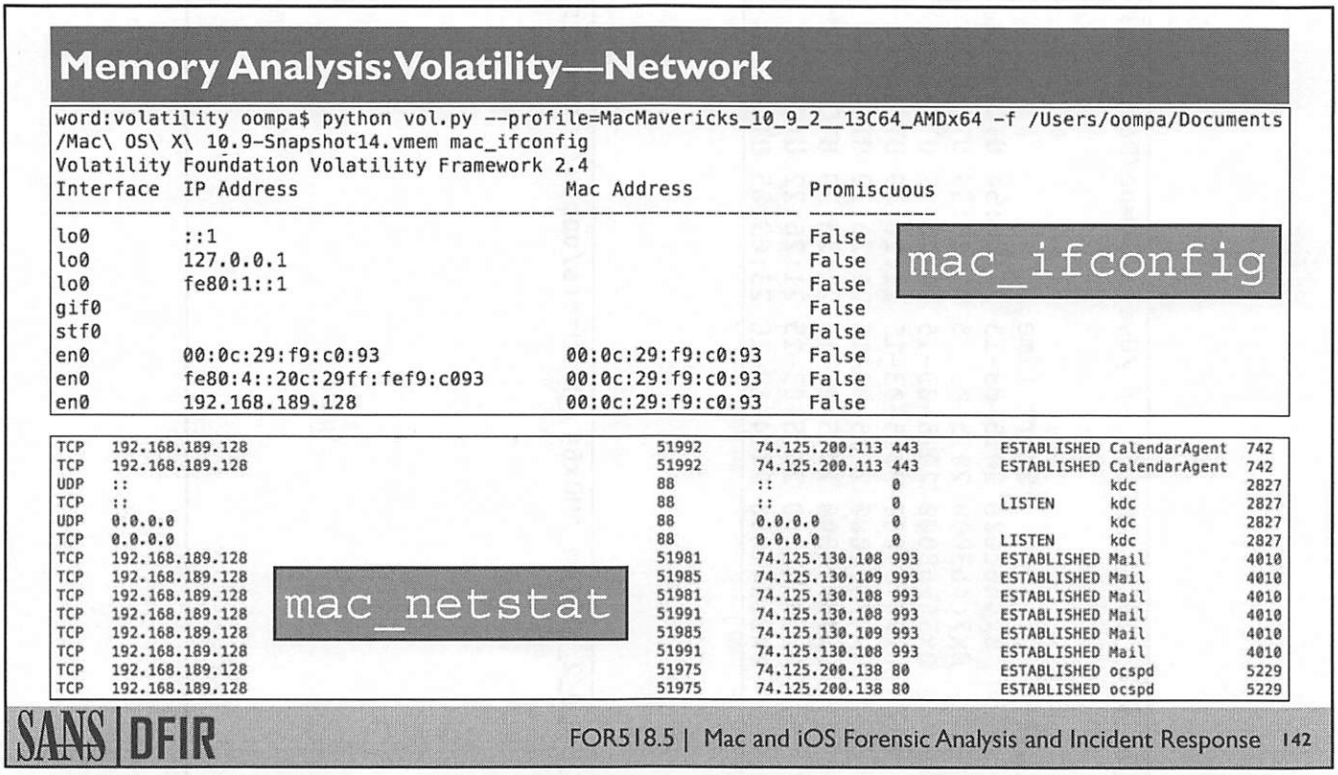
- Process Name
- Process ID (PID)
- UID/GID
- Process Architecture
- Process Start Time

The bottom screenshot shows the output of the `mac_pstree` plugin. This plugin shows the process name, PID, and UID in a tree-like format. This format allows child processes to be determined more quickly.

Reference:
`https://github.com/volatilityfoundation/volatility/wiki/Mac-Command-Reference`

```
word:volatility oompa$ python vol.py --profile=MacMavericks_10_9_2__13C64_AMDx64 -f /Users/oompa/Documents
/Mac\ OS\ X\ 10.9-Snapshot14.vmem mac_pslist
Volatility Foundation Volatility Framework 2.4
Offset (V)         Name             PID  Uid Gid PGID Bits  DTB              Start time
0xffffff80337a74a8 fontworker       5844 501  20 5844 64BIT       0x4b6c000 2015-03-15 21:26:54 UTC+0000
0xffffff803581a5d8 timezoned        5832 210 210 5832 64BIT      0x7c4b3000 2015-03-15 21:26:34 UTC+0000
0xffffff8036ad0000 AirPort Base Sta 5813 501  20 5813 64BIT      0x55e63000 2015-03-15 21:26:29 UTC+0000
0xffffff80350e0df8 rpcsvchost       5806   0   0 5806 64BIT      0x74fe7000 2015-03-15 21:26:29 UTC+0000
0xffffff80337aa7e0 netbiosd         5805 222 222 5805 64BIT      0x47561000 2015-03-15 21:26:29 UTC+0000
0xffffff8035817748 digest-service   5804   0   0 5804 64BIT      0x7d588000 2015-03-15 21:26:29 UTC+0000
0xffffff8036ad3338 syncdefaultsd    5682 501  20 5682 64BIT      0x6d5c4000 2015-03-15 21:26:22 UTC+0000
0xffffff803b662bf0 mdworker         5667 501  20 5667 64BIT      0x3bd41000 2014-12-18 23:03:05 UTC+0000
```

```
word:volatility oompa$ python vol.py --profile=MacMavericks_10_9_2__13C64_AMDx64 -f /Users/oompa/Documents
/Mac\ OS\ X\ 10.9-Snapshot14.vmem mac_pstree
Volatility Foundation Volatility Framework 2.4
Name                 Pid             Uid
kernel_task          0               0
.launchd             1               0
..timezoned          5832            210
..rpcsvchost         5806            0
..netbiosd           5805            222
..digest-service     5804            0
..ocspd              5229            0
..com.apple.WebKit   4028            501
```



Volatility can also show us volatile network information.

The `mac_ifconfig` provides the current IP addresses and network interfaces of the system when the memory dump was captured.

The `mac_netstat` command output includes network connection IP, ports, status, and process affiliation.

Reference:
`https://github.com/volatilityfoundation/volatility/wiki/Mac-Command-Reference`

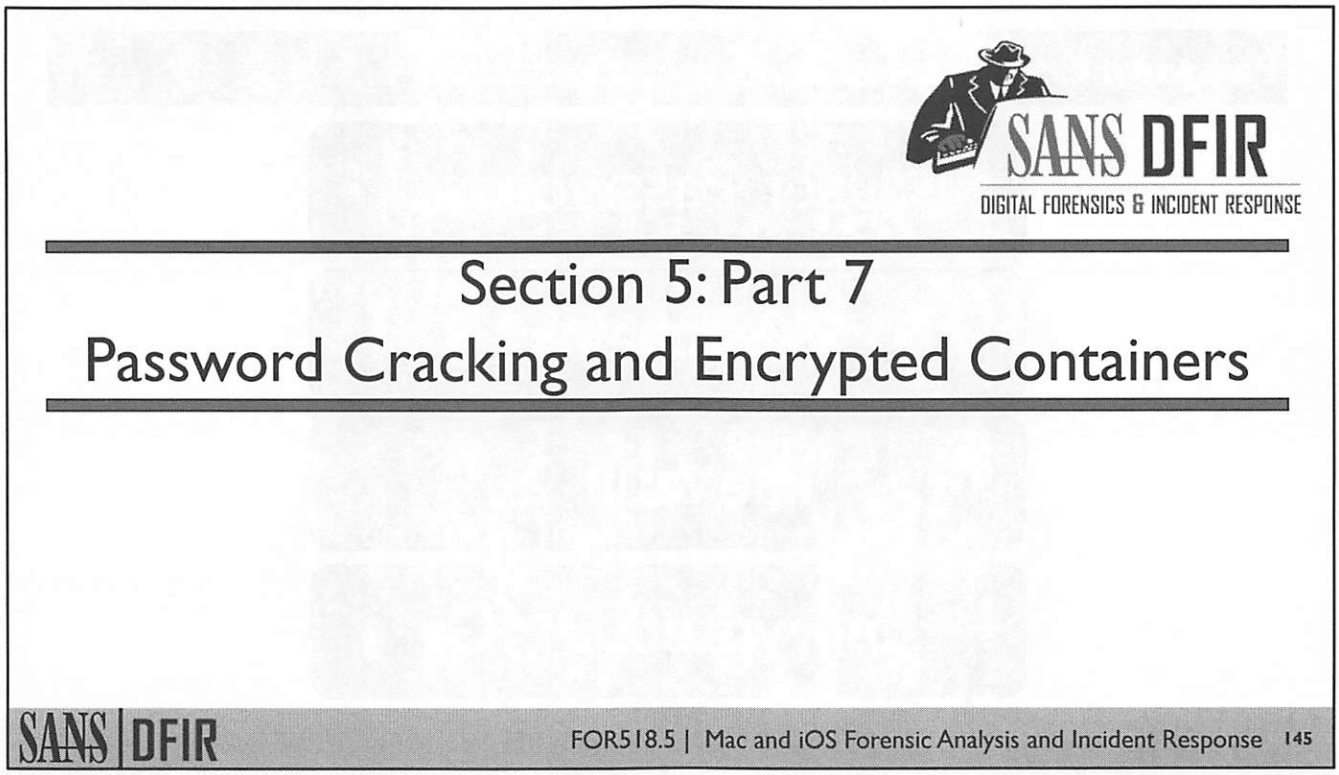
```
word:volatility oompa$ python vol.py --profile=MacMavericks_10_9_2__13C64_AMDx64 -f /Users/oompa/Documents
/Mac\ OS\ X\ 10.9-Snapshot14.vmem mac_ifconfig
Volatility Foundation Volatility Framework 2.4
Interface  IP Address                       Mac Address           Promiscuous
---------- -------------------------------- -------------------- -----------
lo0        ::1                                                   False
lo0        127.0.0.1                                             False
lo0        fe80:1::1                                             False
gif0                                                             False
stf0                                                             False
en0        00:0c:29:f9:c0:93                00:0c:29:f9:c0:93    False
en0        fe80:4::20c:29ff:fef9:c093       00:0c:29:f9:c0:93    False
en0        192.168.189.128                  00:0c:29:f9:c0:93    False
```

```
TCP   192.168.189.128                                51992       74.125.200.113 443         ESTABLISHED CalendarAgent  742
TCP   192.168.189.128                                51992       74.125.200.113 443         ESTABLISHED CalendarAgent  742
UDP   ::                                             88          ::             0                       kdc            2827
TCP   ::                                             88          ::             0           LISTEN      kdc            2827
UDP   0.0.0.0                                        88          0.0.0.0        0                       kdc            2827
TCP   0.0.0.0                                        88          0.0.0.0        0           LISTEN      kdc            2827
TCP   192.168.189.128                                51981       74.125.130.108 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51985       74.125.130.109 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51981       74.125.130.108 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51991       74.125.130.108 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51985       74.125.130.109 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51991       74.125.130.108 993         ESTABLISHED Mail           4010
TCP   192.168.189.128                                51975       74.125.200.138 80          ESTABLISHED ocspd          5229
TCP   192.168.189.128                                51975       74.125.200.138 80          ESTABLISHED ocspd          5229
```

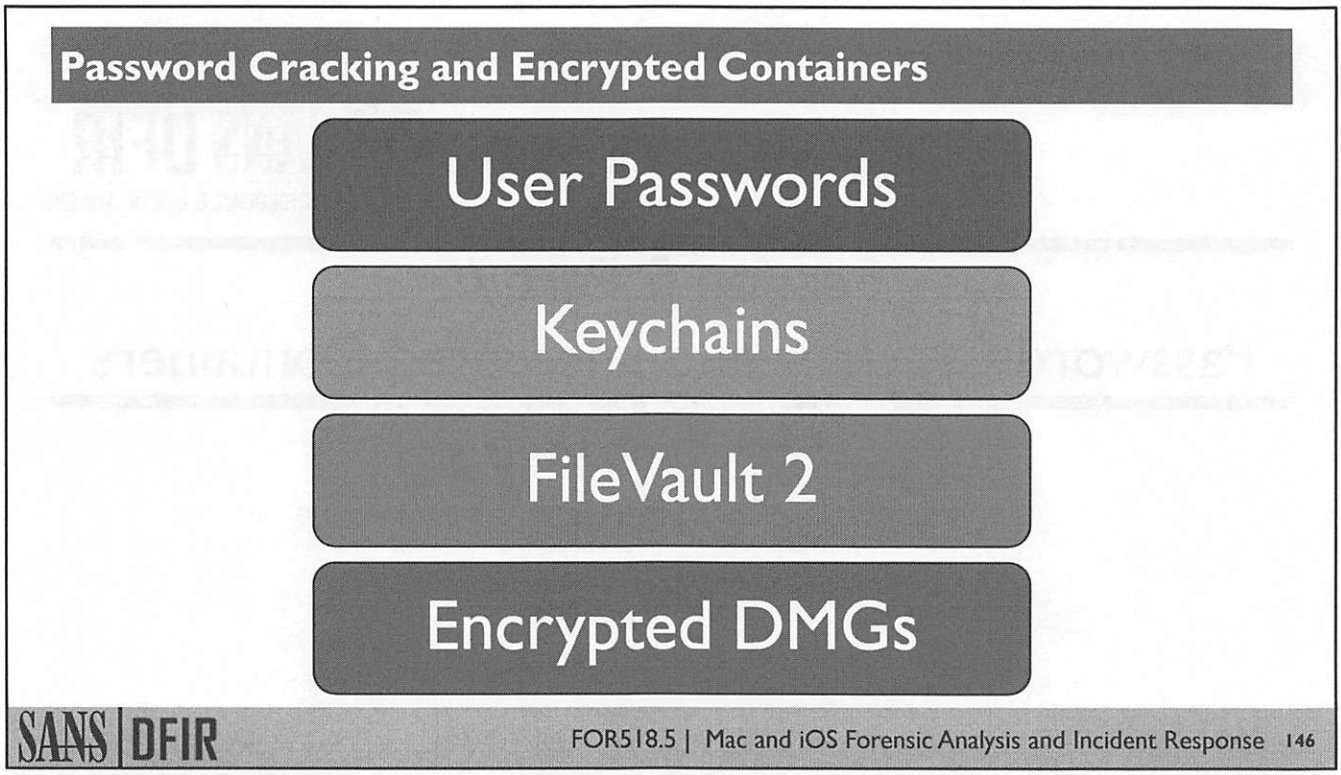
Section 5: Agenda

- Part 1: Pattern of Life
- Part 2: Document Versions
- Part 3: iCloud
- Part 4: Malware and Intrusion Analysis
- Part 5: Live Response
- Part 6: Memory Acquisition and Analysis
- Part 7: Password Cracking and Encrypted Containers

This page intentionally left blank.

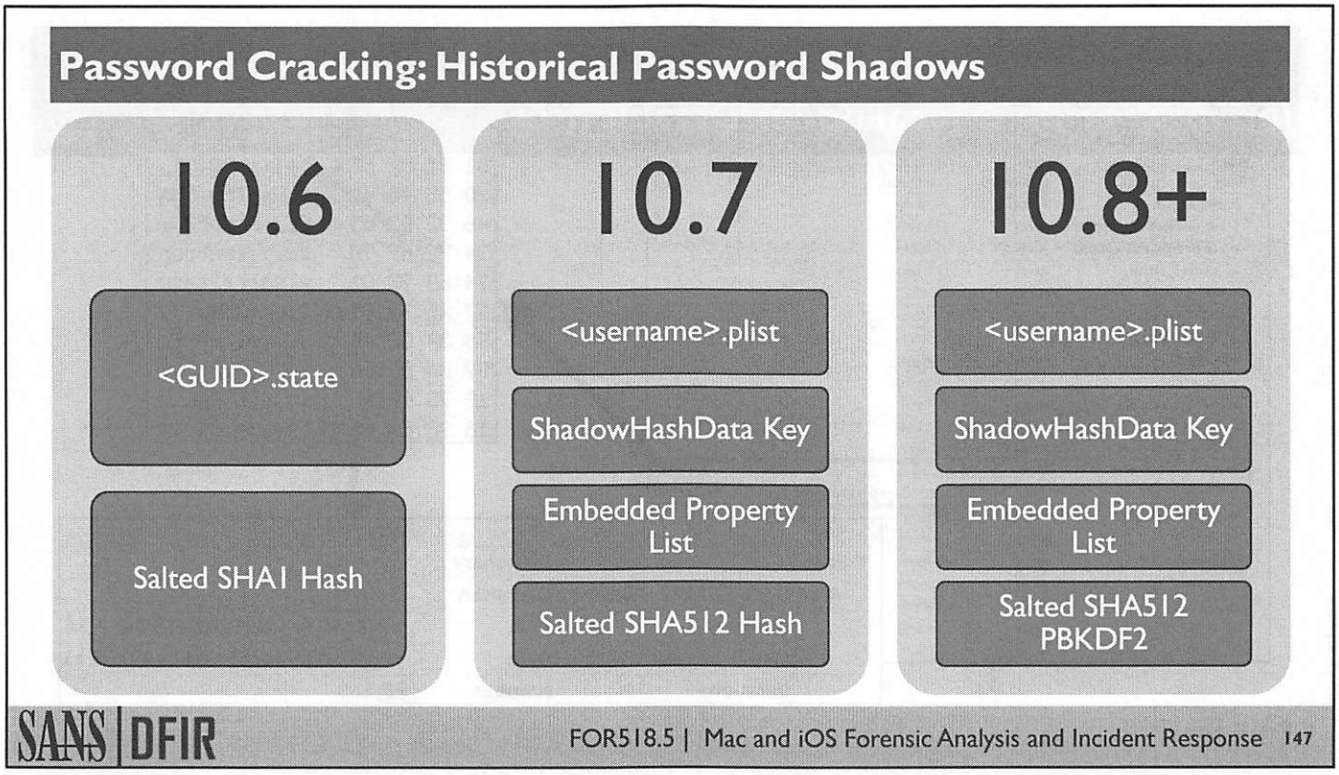


This page intentionally left blank.



An analyst might run into various encrypted containers during their analysis, such as FileVault volumes and encrypted disk image files.

These containers may be able to be cracked open by brute forcing the user passwords or acquiring the passwords from the user's keychain files.



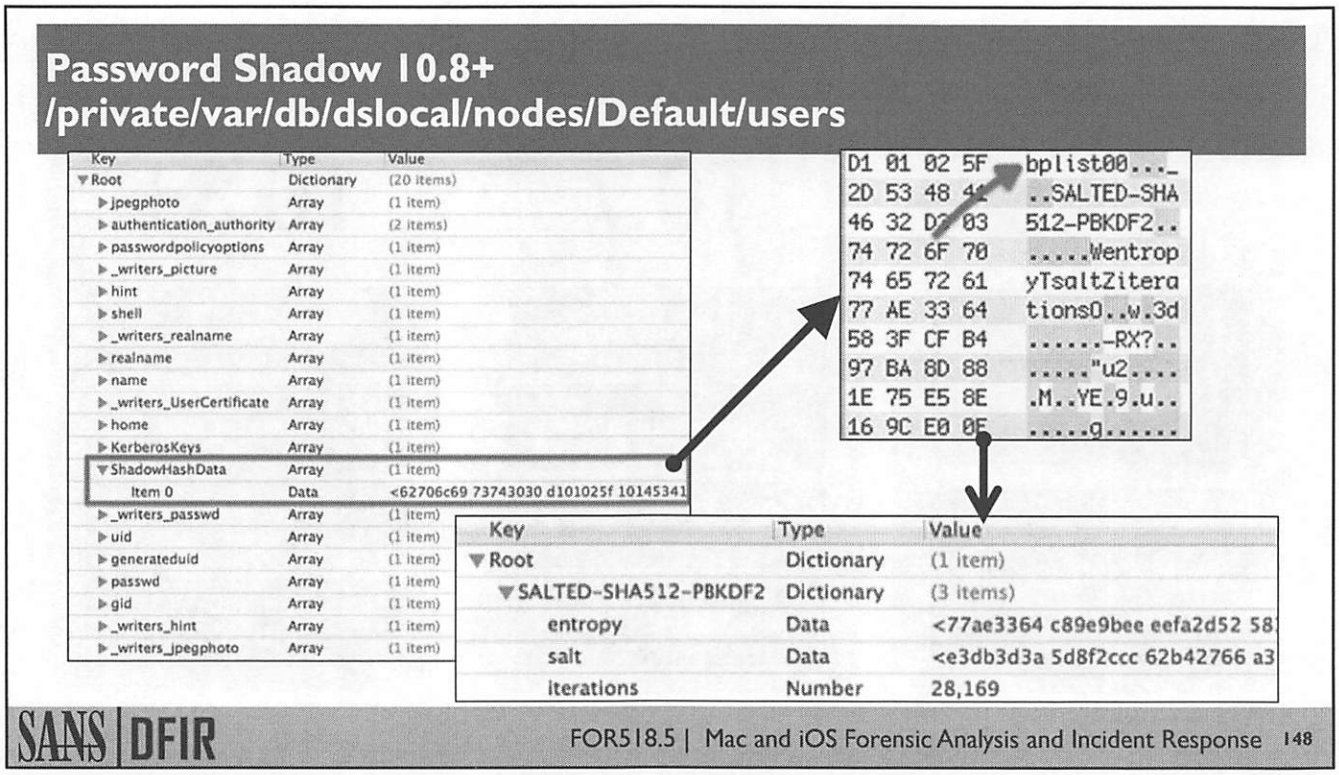
Like any Unix-based system, the user's password is stored in a password shadow file.

In 10.6, the location of the hash is stored in a `<GUID>.state` file located in the `/private/var/db/shadow/hash/` directory. This hash uses a salted SHA1 hash.

In 10.7+ systems, the hash is now located in an embedded property list in the `ShadowHashData` key in the user property list located in the `/private/var/db/dslocal/nodes/Default/users/` directory.

10.7 systems use a salted SHA512 hash, while 10.8+ systems use a salted SHA512 PBKDF2 hash.

The hashing algorithms have been increasingly more difficult to crack with every version of OS X. (We will see a good example of this in the related exercise!)



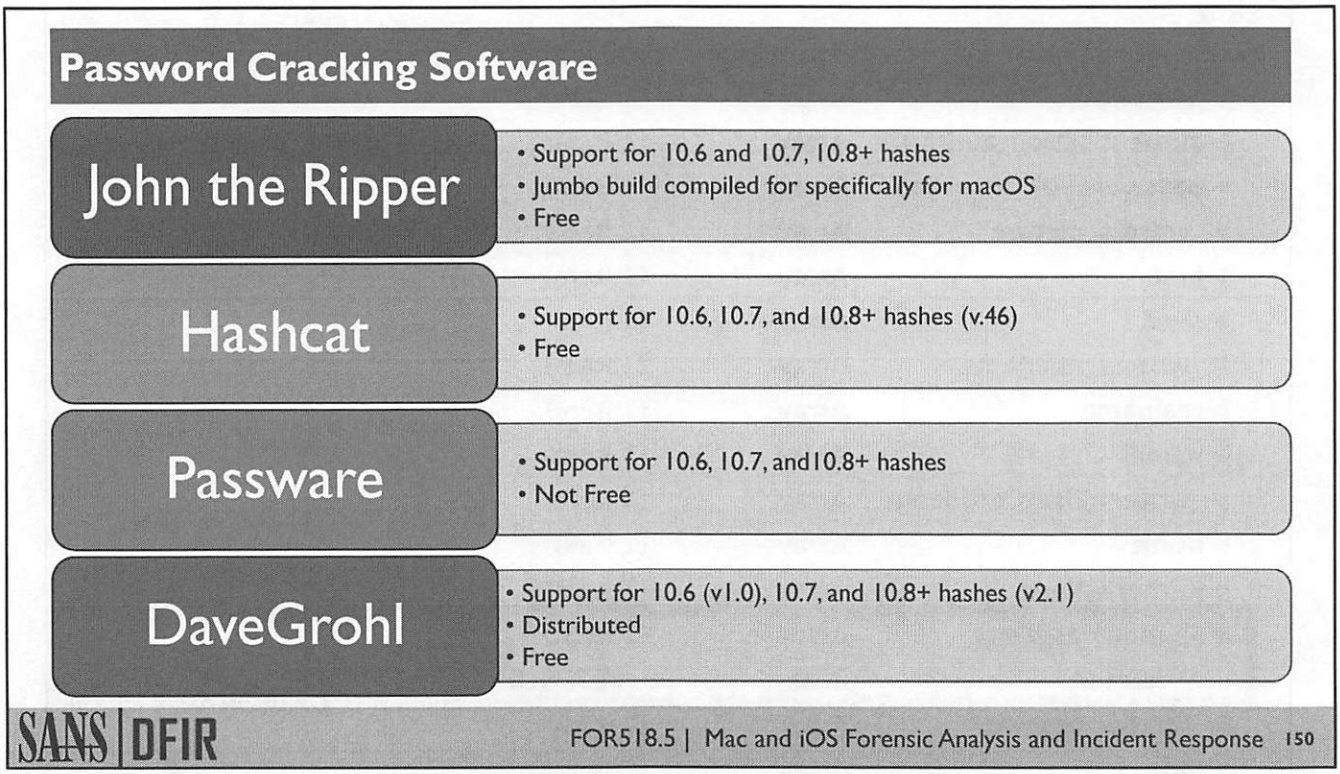
Similar to 10.7, 10.8+ systems store the hash in an embedded property list within the `ShadowHashData` key within the user's property list file located in the `/private/var/db/dslocal/nodes/Default/users/` directory.

This property list can be extracted and viewed. We can see the keys for entropy, salt, and iterations. The hash algorithm differs from 10.7, as it uses the SHA512 PBKDF2 algorithm.

Key	Type	Value
▼ Root	Dictionary	(20 items)
▶ jpegphoto	Array	(1 item)
▶ authentication_authority	Array	(2 items)
▶ passwordpolicyoptions	Array	(1 item)
▶ _writers_picture	Array	(1 item)
▶ hint	Array	(1 item)
▶ shell	Array	(1 item)
▶ _writers_realname	Array	(1 item)
▶ realname	Array	(1 item)
▶ name	Array	(1 item)
▶ _writers_UserCertificate	Array	(1 item)
▶ home	Array	(1 item)
▶ KerberosKeys	Array	(1 item)
▼ ShadowHashData	Array	(1 item)
Item 0	Data	<62706c69 73743030 d101025f 10145341
▶ _writers_passwd	Array	(1 item)
▶ uid	Array	(1 item)
▶ generateduid	Array	(1 item)

```
D1 01 02 5F   bplist00..._
2D 53 48 41   ..SALTED-SHA
46 32 D2 03   512-PBKDF2..
74 72 6F 70   .....Wentrop
74 65 72 61   yTsaltZitera
77 AE 33 64   tionsO..w.3d
58 3F CF B4   ......-RX?..
97 BA 8D 88   ....."u2....
1E 75 E5 8E   .M..YE.9.u..
16 9C E0 0E   .....g......
```

Key	Type	Value
▼ Root	Dictionary	(1 item)
▼ SALTED-SHA512-PBKDF2	Dictionary	(3 items)
entropy	Data	<77ae3364 c89e9bee eefa2d52 58
salt	Data	<e3db3d3a 5d8f2ccc 62b42766 a3
iterations	Number	28,169



Various password cracking software is available. These programs range from free to quite expensive. The software programs listed above work for each of the different password hashing algorithms used in 10.6–10.8+.

References:
DaveGrohl: `http://www.davegrohl.org/`
John the Ripper:

- `http://openwall.info/wiki/john/custom-builds#Compiled-for-Mac-OS-X`
- `http://download.openwall.net/pub/projects/john/contrib/macosx/`

Hashcat: `http://hashcat.net/hashcat/`
Passware: `http://www.lostpassword.com/kit-forensic/`

Cracking Keychains

Dump the unlocked login.keychain-db on logged-on system

- `security dump-keychain -d`

Acquire login password

- `security unlock-keychain -p <password> <keychain>`

John the Ripper: keychain2john

- `./keychain2john login.keychain-db > login_keychain.txt`
- `./john login_keychain.txt`

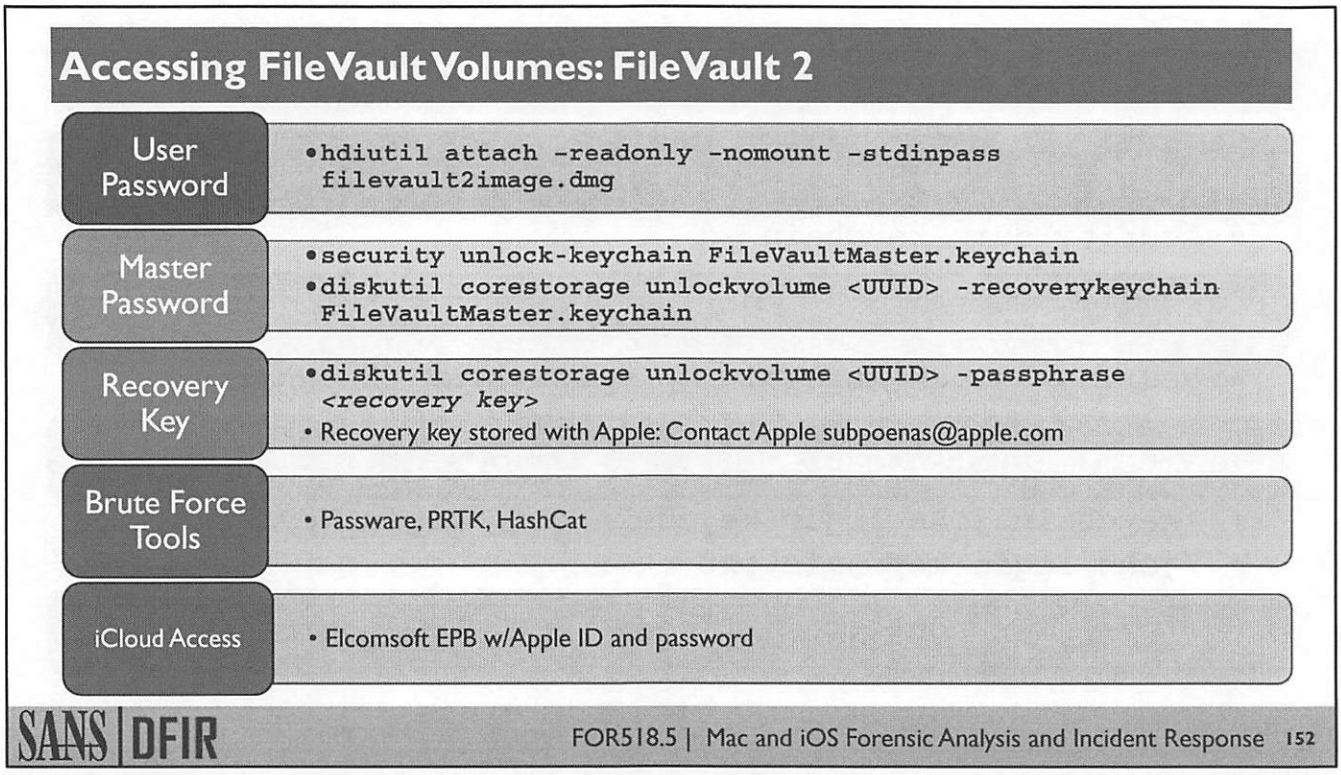
Passware: Optimized but not free

Keychains can be useful to gather a variety of passwords. Many people reuse passwords or parts of passwords. This can be used to shorten the wordlist to brute force other passwords.

On an already logged-on system, an investigator is able to dump the passwords (usually without a password) of the currently logged-on user's `login.keychain-db`. The `security` command used with the `dump-keychain -d` option can be used to do this. This command produces quite a bit of output, so redirection to a file is recommended for later viewing.

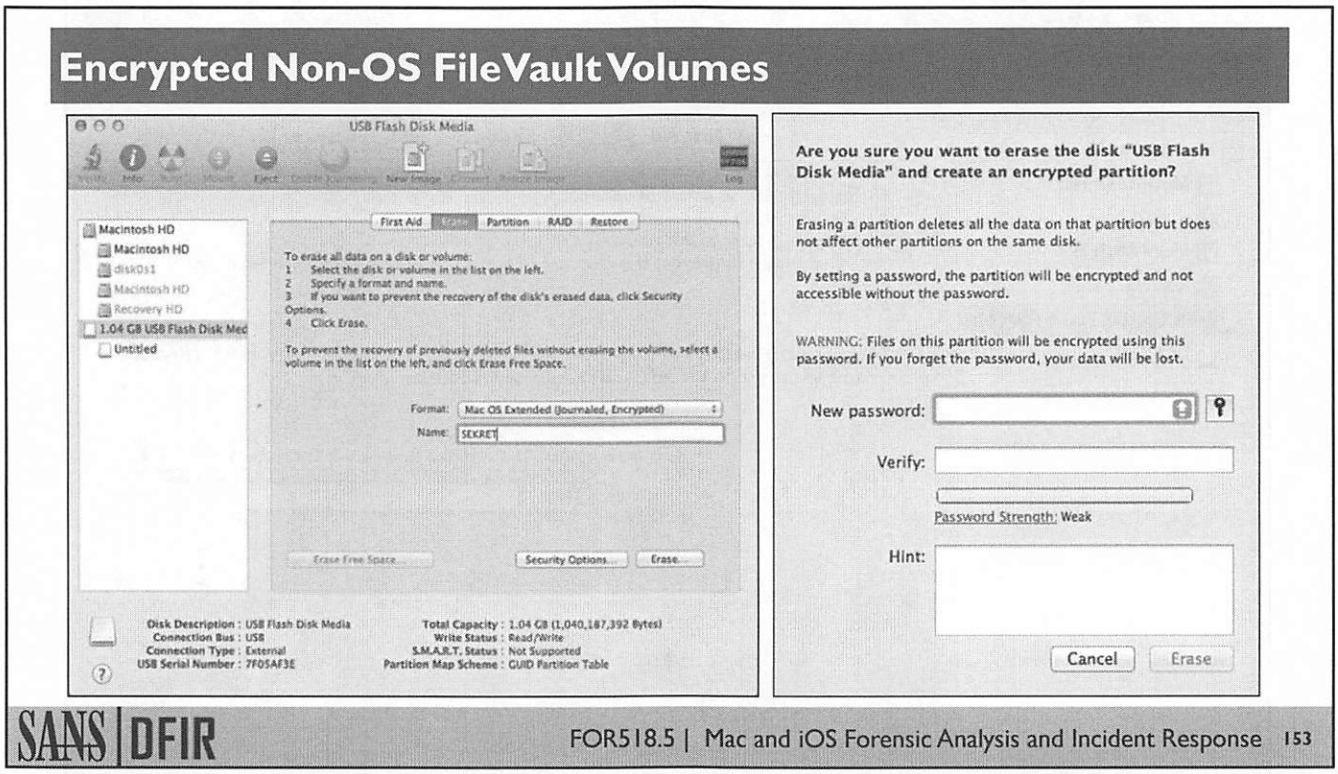
If a user's logon password can be acquired, an analyst can usually use that to access the keychain by using the `unlock-keychain` option to the `security` command—"usually" being a key term. The user's password is often used as the default password to a user's keychain; however, a user can change this at any time.

The well-known password cracking program John the Ripper can be used to brute force keychain passwords. Specially compiled versions (at least 1.7.9 Jumbo 6) of JTR will have the program `keychain2john` that is used to dump the password hash from the keychain file and into JTR format. This output can be saved to a file (`login_keychain.txt` for example) and used with the John program. It should be noted that this may take a LONG time to crack and highly depends on the configuration, JTR wordlists, and the complexity of the password. A good resource for compiling and configuring JTR for this purpose can be found here: `http://easymactips.blogspot.com/2012/09/john-ripper-tutorial-examples-and.html`



FileVault 2 volumes can be accessed by using the user's password, the master password, or a recovery key, or by a brute force program like Passware/PRTK/HashCat. If the user selects to store/reset their system password using their iCloud accounts, that is one more avenue to get into their FileVault disks.

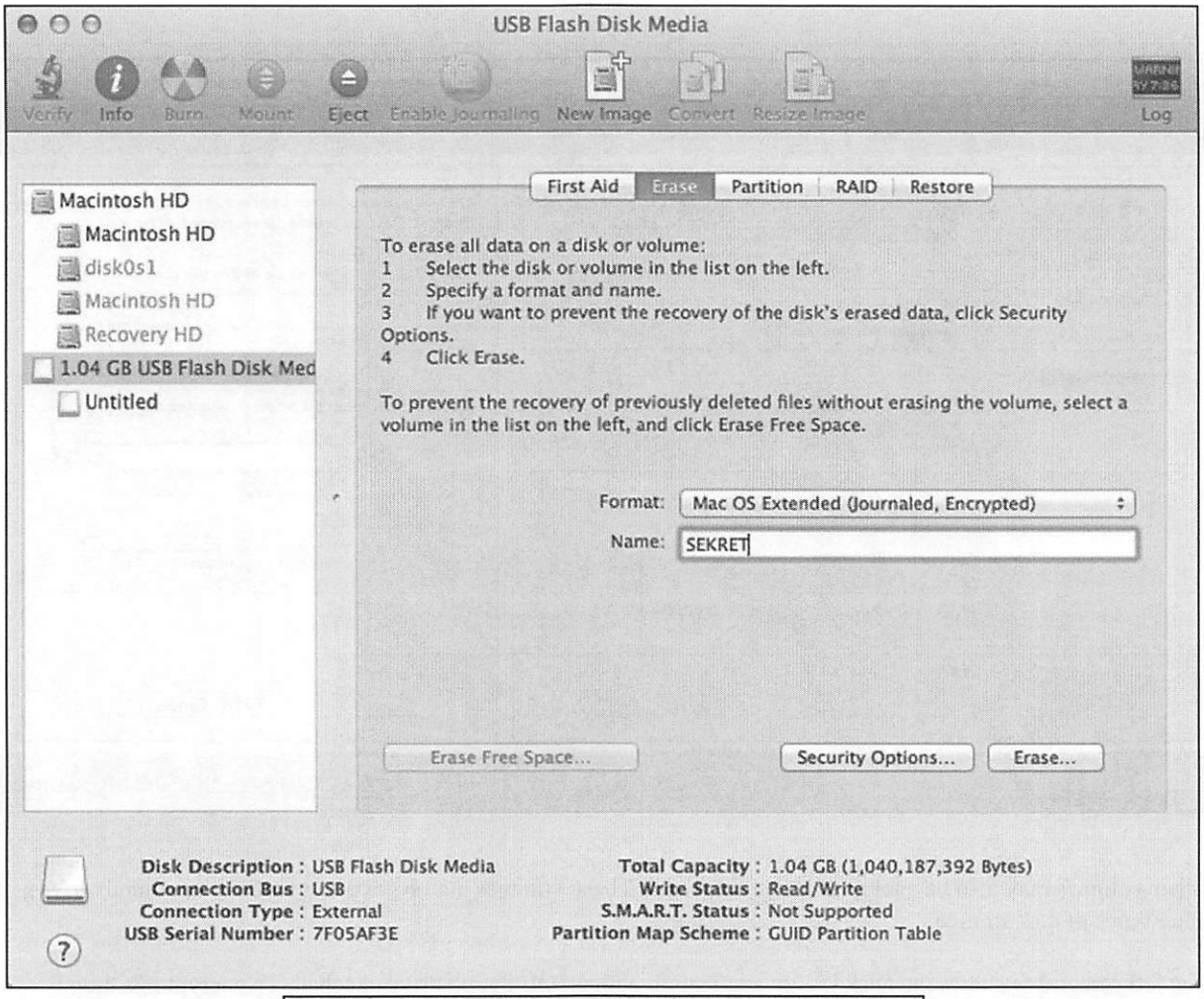
References:
`http://derflounder.wordpress.com/2011/11/23/using-the-command-line-to-unlock-or-decrypt-your-filevault-2-encrypted-boot-drive/`
`https://support.passware.com/hc/en-us/articles/115002145727-How-to-decrypt-Full-Disk-Encryption`
`https://support.apple.com/en-us/HT204837`
`https://blog.elcomsoft.com/2016/08/breaking-filevault-2-encryption/`
`https://blog.passware.com/2018/10/03/passware-kit-2018-v2/`

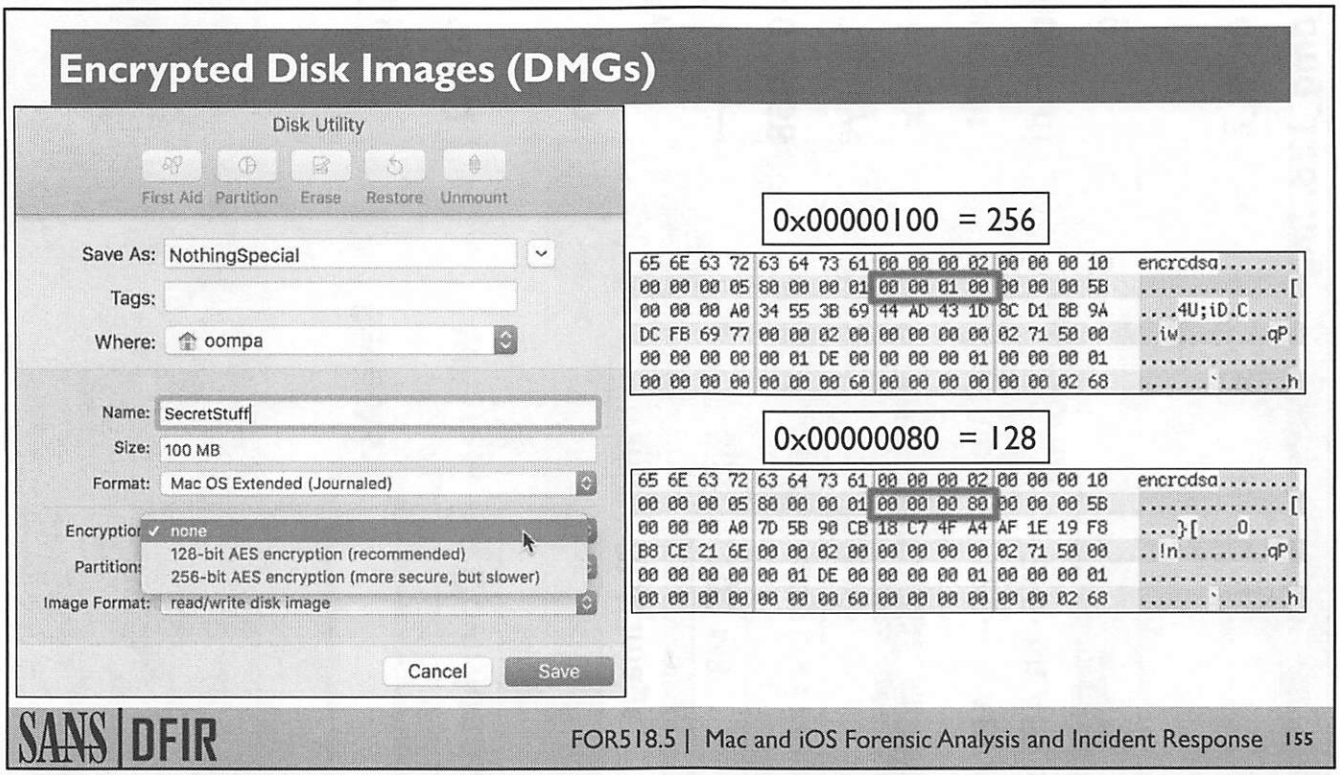


Other volumes may also be encrypted using FileVault. These volumes may be external hard drives, thumb drives, or other types of disk storage.

The left screenshot shows the Disk Utility application with a 1GB thumb drive attached. To encrypt this thumb drive, a user will need to go to the "`Erase`" tab and choose "`Mac OS Extended (Journaled, Encrypted)`" in the Format options.

The right screenshot shows the pop-up window the user will receive when the "`Erase...`" procedure is started. This window allows the user to input a password (and password hint) for the volume.





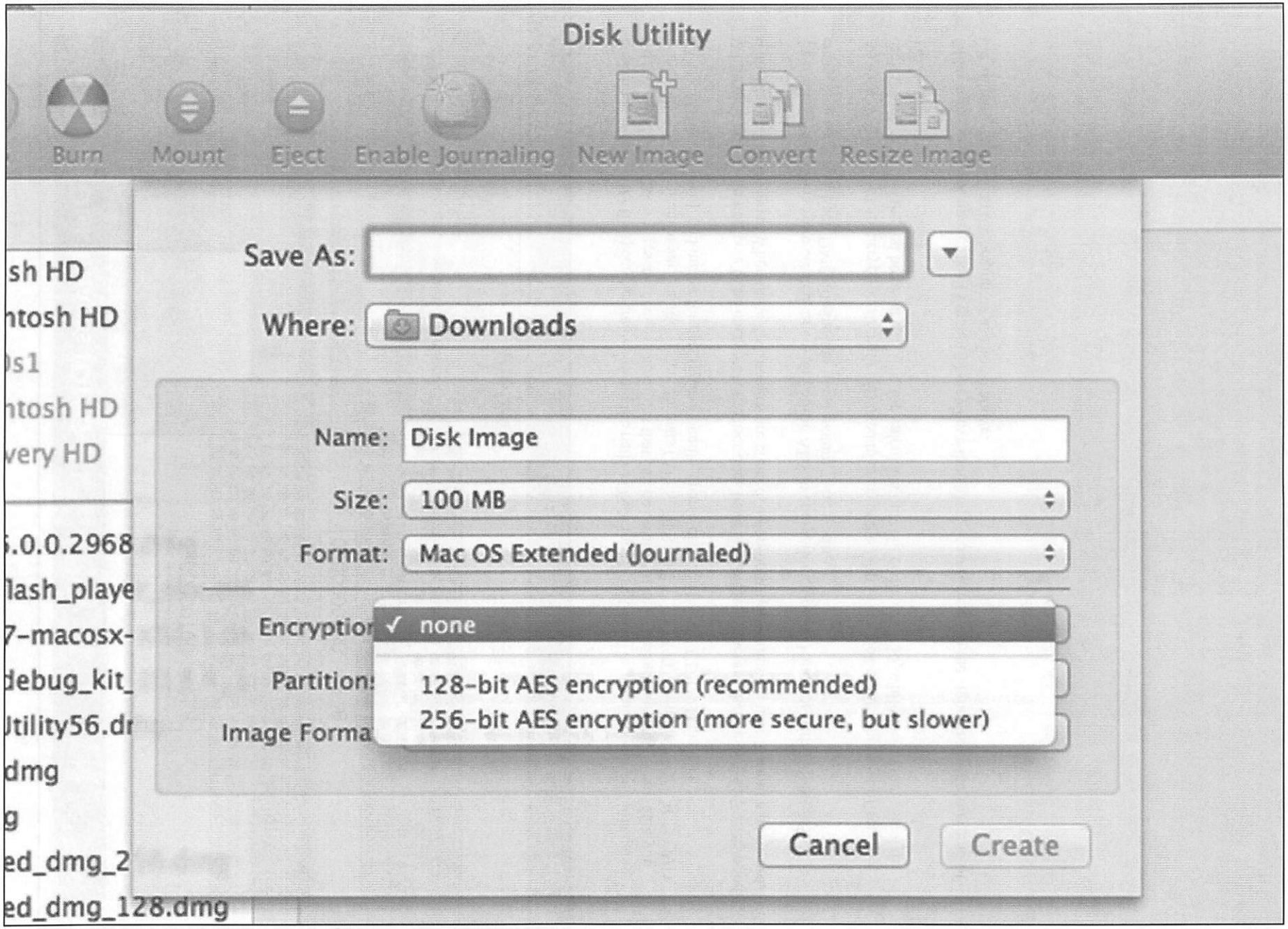
Disk Images (or DMG files) can be created using the native OS X tool Disk Utility or via the command line using `hdiutil`. These disk images are encapsulated files that can be created in any size and can contain any data you want. Other options include various file system formats (APFS, HFS+, FAT, ExFAT), partition choices (CD/DVD, GUID, MBR, or None), and image formats (sparse bundle, sparse disk, read/write, CD/DVD master).

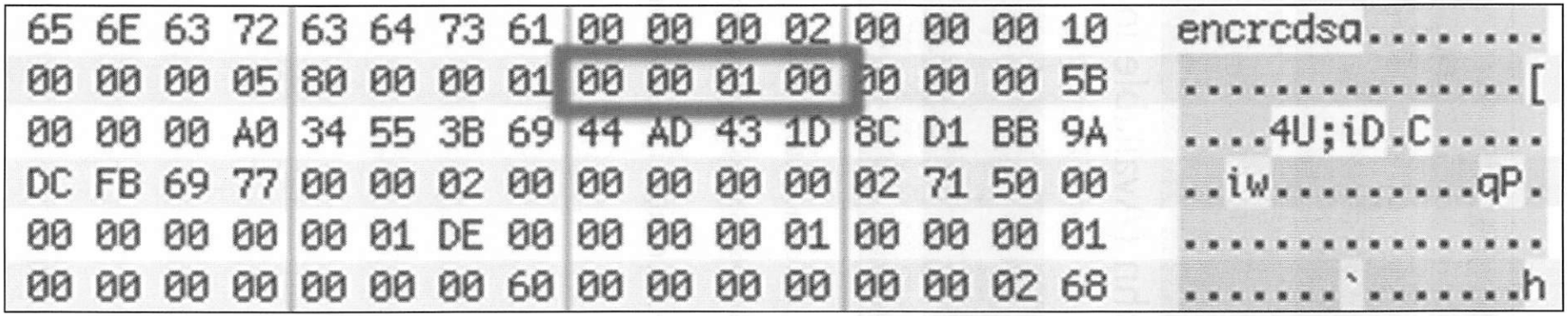
While disk images do not have to be encrypted, the user does have the ability to encrypt them using AES encryption with 128- or 256-bit keys. When these disk images are created, a pop-up window allows a user to input a password.

Two encrypted DMGs were created, one with 256-bit AES in encryption (top), and another with 128-bit AES encryption (bottom). Each DMG file has the file signature "`encrcdsa`".

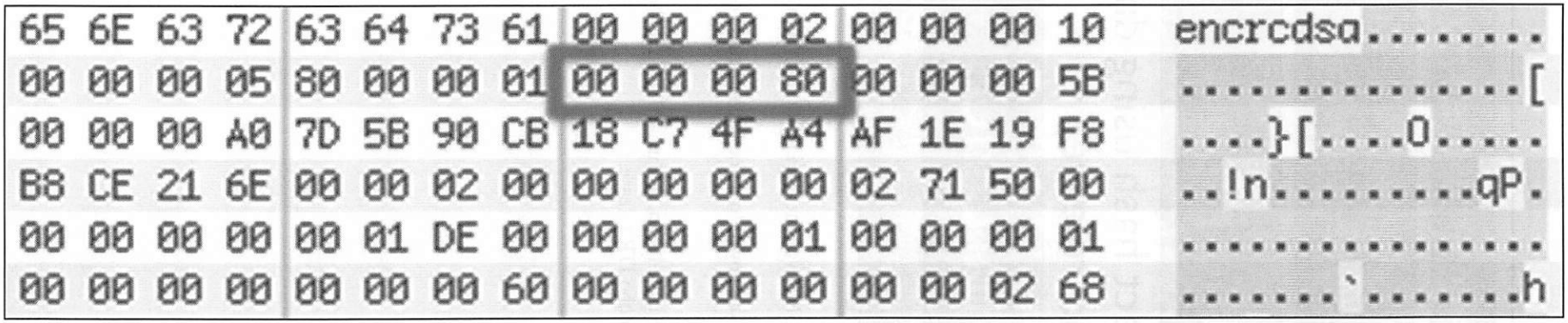
An easy way to tell what encryption strength was used to encrypt the DMG is to look at the 4 bytes at offset `24`; each is highlighted in the red boxes above. The top calculates out (big endian) to 256, while the bottom calculates to 128.

Older `sparsedisk` image files (Version 1) may have the string "`cdsaencr`" in the signature at the very end of the volume, rather than the beginning as shown above.

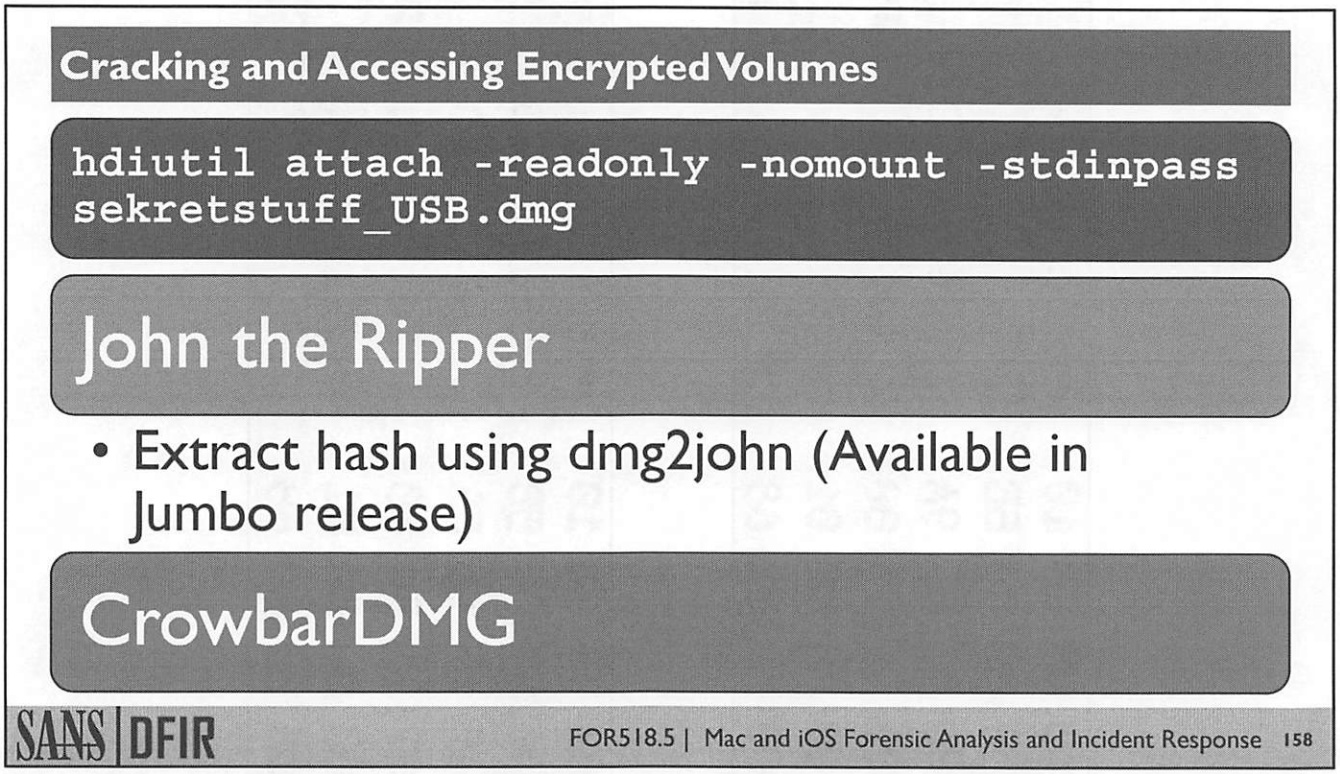




```
65 6E 63 72 63 64 73 61 00 00 00 02 00 00 00 10   encrcdsa........
00 00 00 05 80 00 00 01 00 00 01 00 00 00 00 5B   ...............[
00 00 00 A0 34 55 3B 69 44 AD 43 1D 8C D1 BB 9A   ....4U;iD.C.....
DC FB 69 77 00 00 02 00 00 00 00 00 02 71 50 00   ..iw.........qP.
00 00 00 00 00 01 DE 00 00 00 00 01 00 00 00 01   ................
00 00 00 00 00 00 00 60 00 00 00 00 00 00 02 68   .......`.......h
```



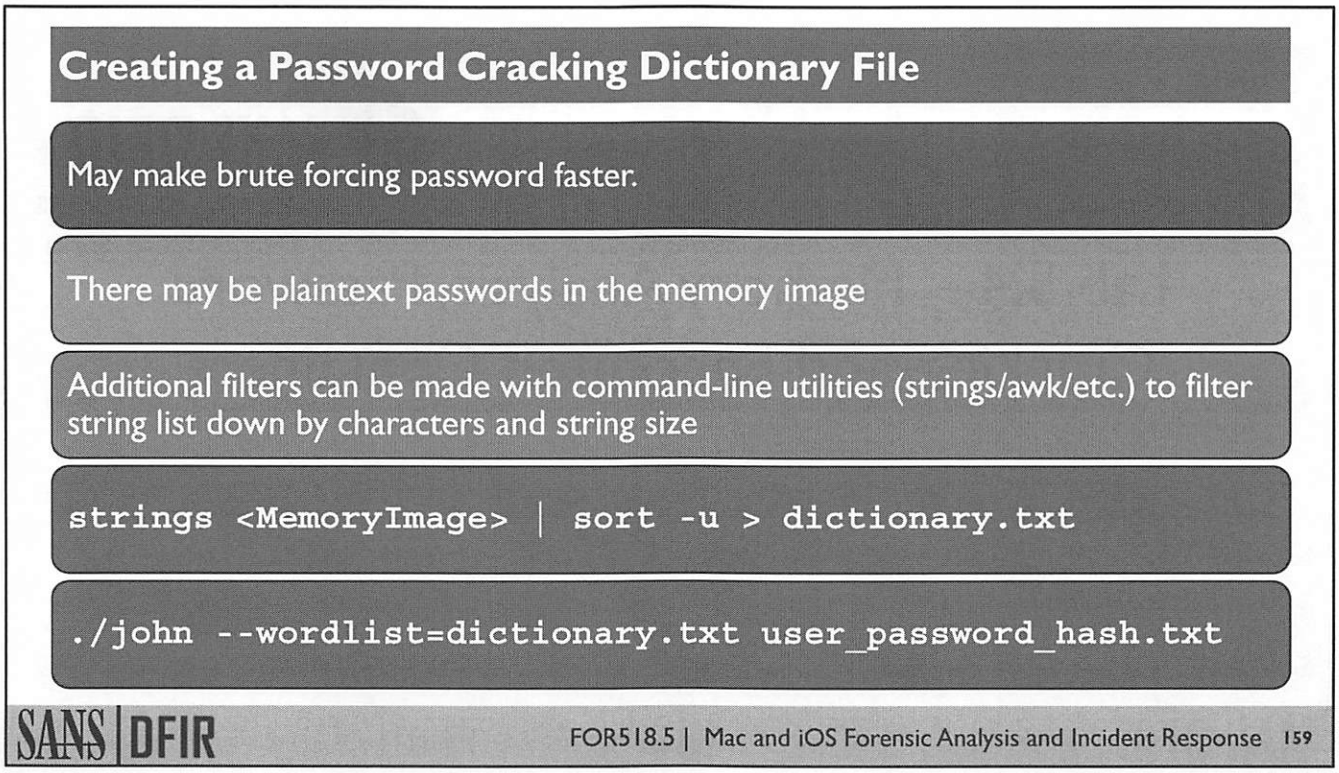
```
65 6E 63 72 63 64 73 61 00 00 00 02 00 00 00 10   encrcdsa........
00 00 00 05 80 00 00 01 00 00 00 80 00 00 00 5B   ...............[
00 00 00 A0 7D 5B 90 CB 18 C7 4F A4 AF 1E 19 F8   ....}[....O.....
B8 CE 21 6E 00 00 02 00 00 00 00 00 02 71 50 00   ..!n.........qP.
00 00 00 00 00 01 DE 00 00 00 00 01 00 00 00 01   ................
00 00 00 00 00 00 00 60 00 00 00 00 00 00 02 68   .......`.......h
```



These encrypted DMG volumes can be mounted using the `hdiutil` command shown above.

If the password is unknown, John the Ripper and CrowbarDMG are able to access and attempt to brute force these volumes.

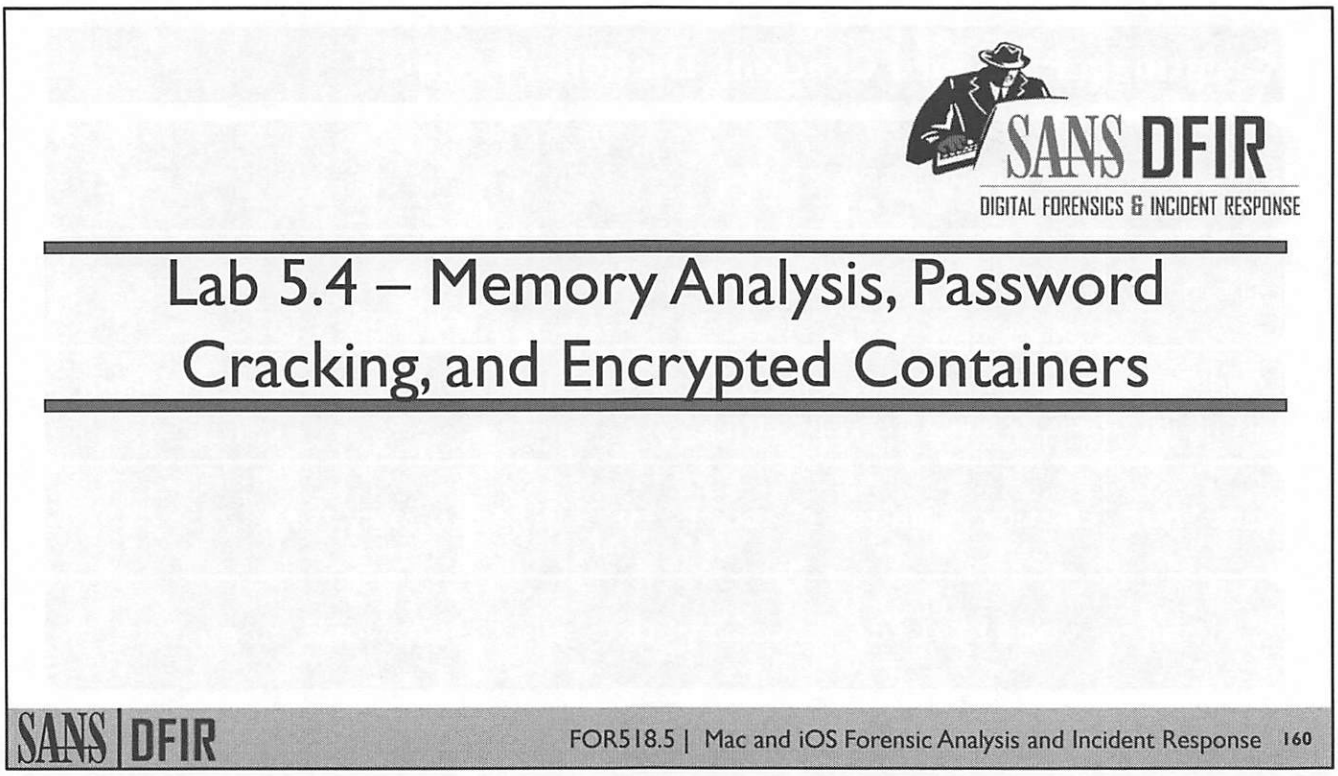
A special program called `dmg2john` will have to be used with John the Ripper to extract the hash. This utility is available in the Jumbo JTR release.



One way to make password brute forcing faster is to use a dictionary file. If you were lucky enough to capture a RAM dump at the time of acquisition, very often plaintext passwords are stored in RAM.

To create a dictionary file, use the `strings` command, uniquely sort the output, and save the contents in a text file to be used with your password cracking software of choice.

```
strings <MemoryImage> | sort -u > dictionary.txt
```



This page intentionally left blank.

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response

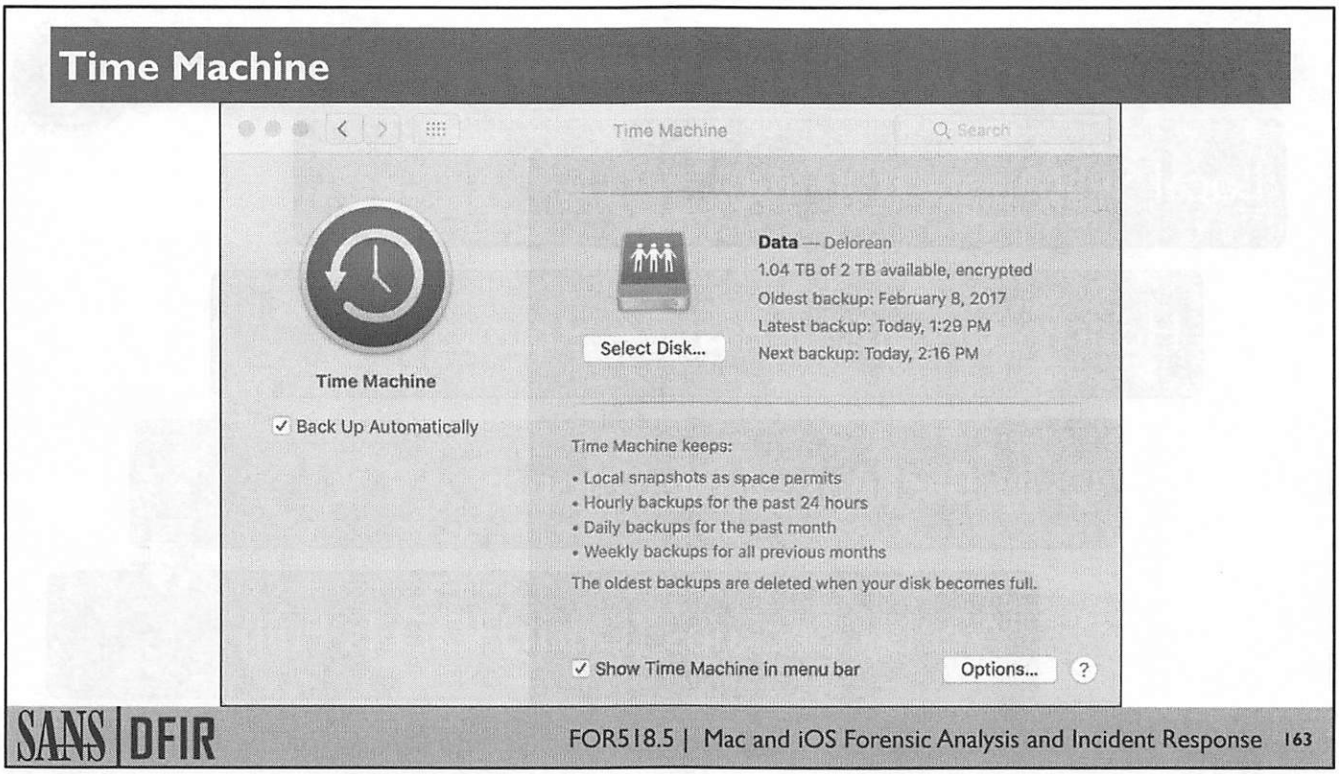
FOR518 Section 5: Advanced Analysis Topics

Author: Sarah Edwards
oompa@csh.rit.edu
mac4n6.com
http://twitter.com/iamevltwin

https://digital-forensics.sans.org/
http://twitter.com/sansforensics

Appendix: Time Machine

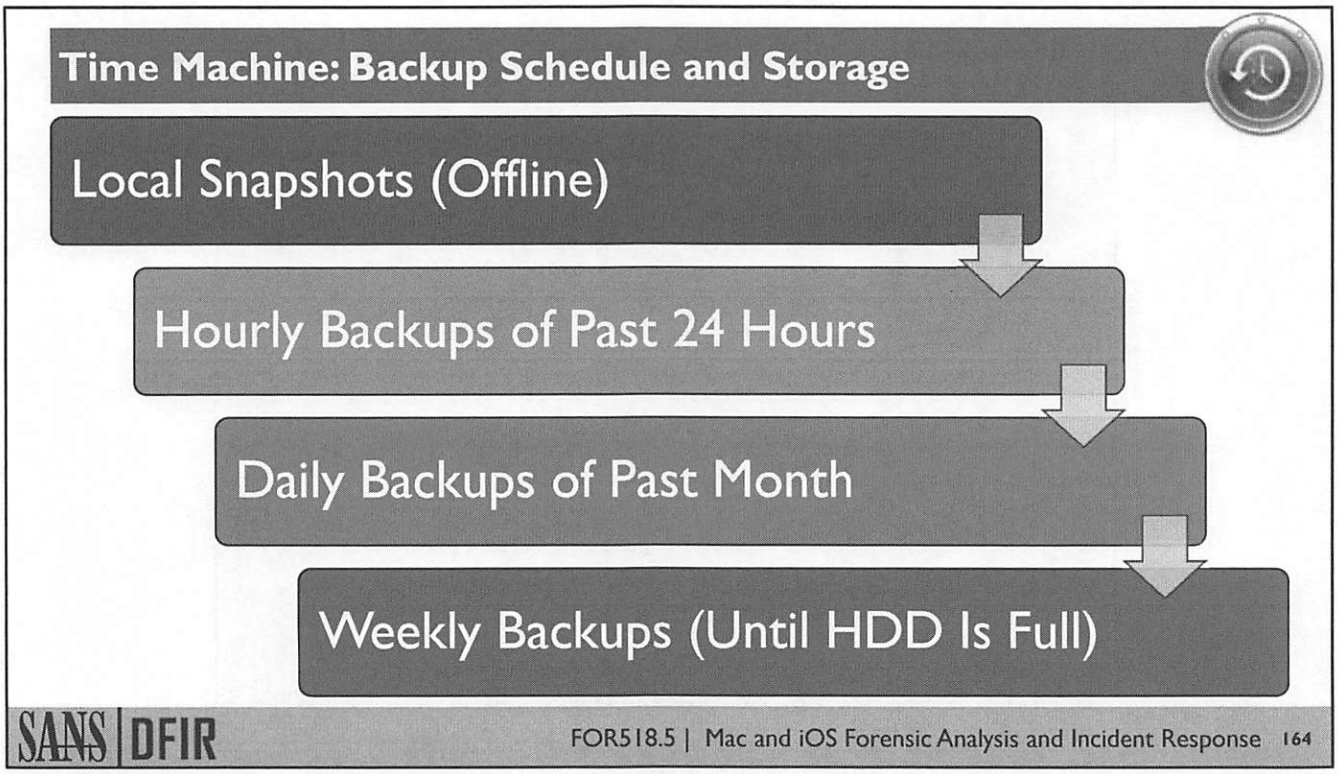
This page intentionally left blank.



The native backup program on OS X is called Time Machine and was introduced in version 10.5. This application can interface with Apple Time Capsule, other network-based storage, or external hard drives.

Whenever the system detects a large external hard drive connected, or accesses a network storage device, a pop-up window will ask the user if they would like to use the disk as a Time Machine backup disk. These disks can be used with automated or manual backup systems, depending on what configurations the user selects.

Shown in the screenshot above, a networked backup device named "`Delorean`" contains a backup partition named "`Data`". This disk is two terabytes in size and has 1.04 terabytes available. The Time Machine window will show data with respect to when the oldest and latest backups were completed and when the next backup is scheduled to take place.



Time Machine creates incremental backups of a system on a default schedule. This schedule can be changed by using the `defaults` command or editing the `com.apple.backupd.plist` property list file.

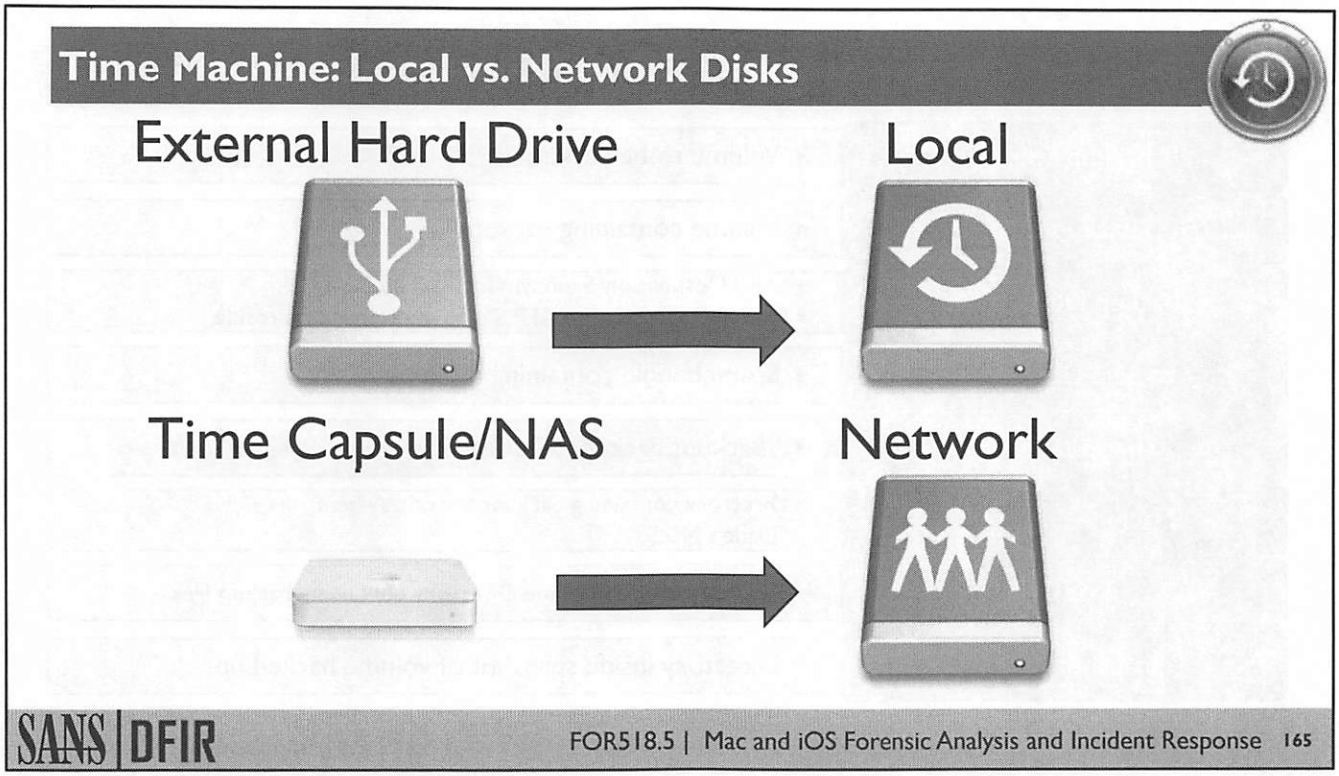
Local snapshots, also called Mobile Backups, are created when the Time Machine volume is not present on the system … such as when a user is traveling and their Time Capsule is at home.

Hourly backups are created and kept for 24 hours. Daily backups are created and kept every day for the past month, and weekly backups are created and saved until the disk runs out of storage space.

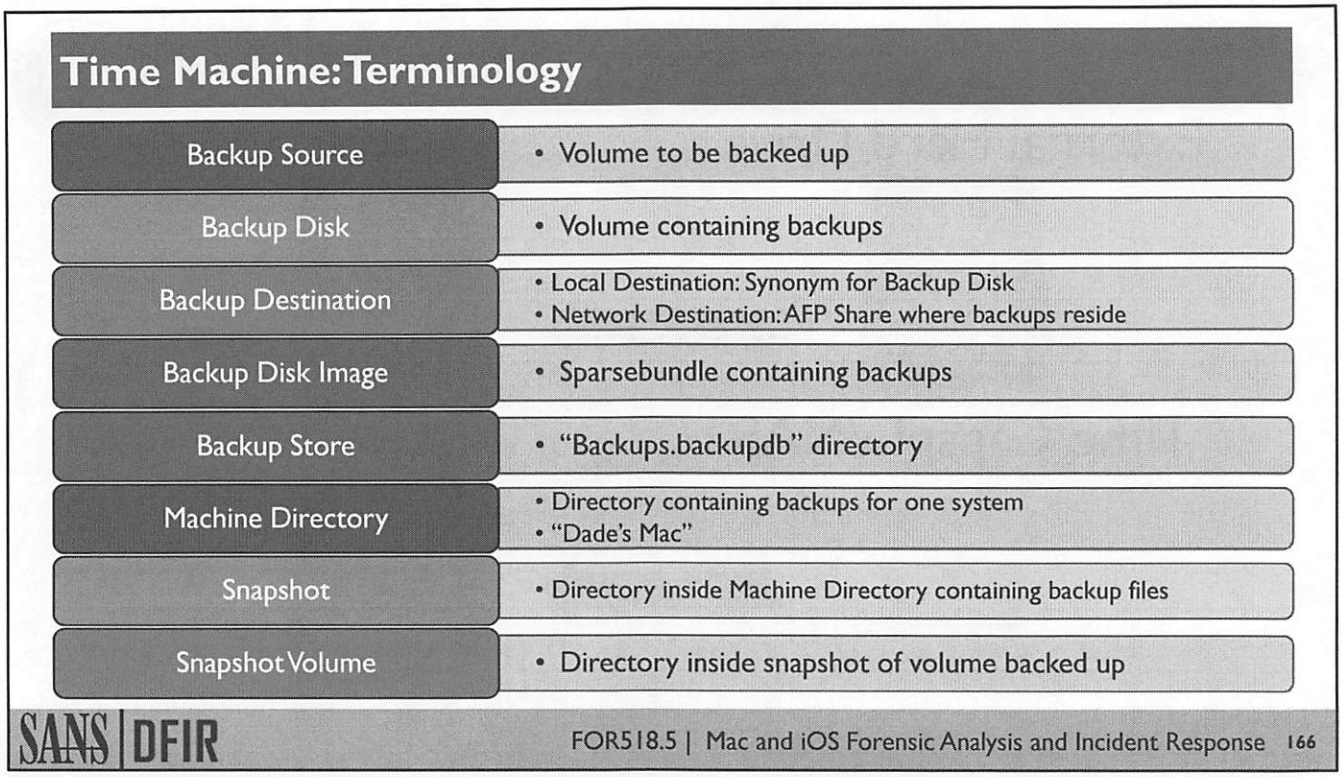
Power Nap is a feature introduced in 10.8 for specific hardware systems (newer MacBook Pros and Airs). This allows the system to continue to create backups, along with other system functions, such as checking for mail and downloading software updates while the Mac is in sleep mode.

References:
Mac Basics: Time Machine
`https://support.apple.com/en-us/HT201250`

Mountain Lion: Power Nap
`https://support.apple.com/en-us/HT204015`

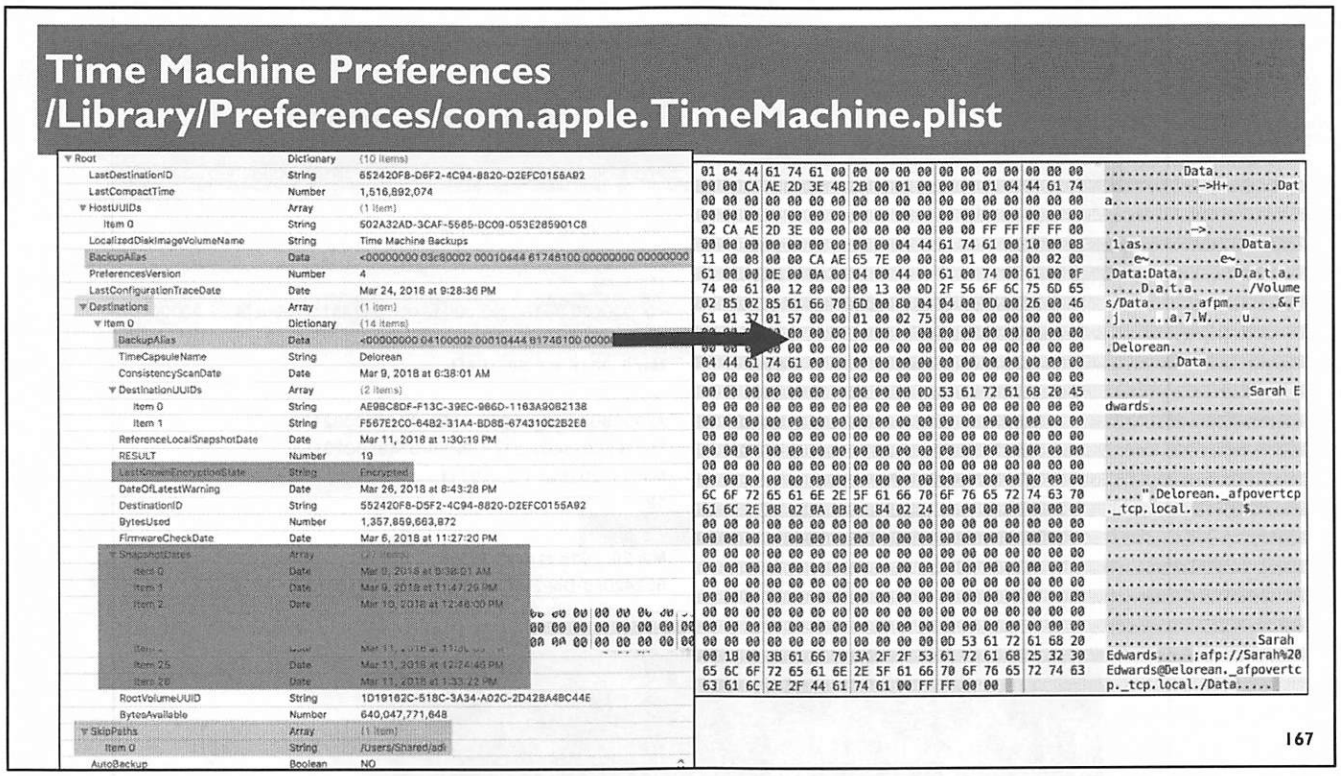


A network backup device has an icon that looks like a blue disk with paper dolls. An external (non-network) device will have an icon that looks like a green disk with a "Time Machine" arrow/clock—similar to the Time Machine application icon shown above.



The Time Machine application uses specific terms to identify certain parts of the backup disk and process.

Reference:
`tmutil` Man Page



The backup data location used with Time Machine is located in the `com.apple.TimeMachine.plist` property list. This property list contains the GUIDs for the Destination Volume (the Time Capsule or external HDD) and the Root Volume (likely named Macintosh HD, the boot volume of the system).

This property list also contains certain Time Machine preference keys, such as:

- `AutoBackup`: Backup automatically (versus manual backups)
- `LastCompactTime`: The date in Unix epoch time that the backup sparse bundle has been "compacted" to create more space. The sparse bundle is unique to network-based Time Machine volumes.
- `SkipPaths`: List of user-configured file paths to explicitly skip on backup
- `IncludeByPath`: List of user-configured file paths to explicitly include on backup
- `BackupAlias`: Binary alias data of the Time Machine backup volume (shown right); this can help us determine if this is a local or network backup. This example is from a network backup. The URL contains the handler "`afp://`". This protocol is the Apple Filing Protocol that is commonly used with Time Machine and the Apple network backup drive, Time Capsule. In the example, we can see that the user creating the backup is "`Sarah Edwards`". The Time Capsule is named "`Delorean`", and the backup data is stored on the "`Data`" partition of the Time Capsule.

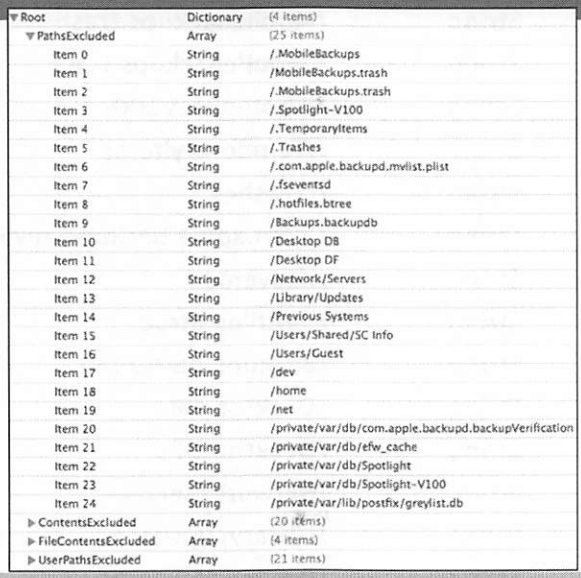
The `Destinations` key contains subkeys for each time the machine backs up a volume. These keys contain information pertaining to the backup volume, such as:

- `BytesUsed`: Bytes allocated by the backups
- `SnapshotDates`: Snapshot timestamps
- `BytesAvailable`: Available space of the backup volume

▼ Root	Dictionary	(10 items)
LastDestinationID	String	552420F8-D5F2-4C94-8820-D2EFC0155A92
LastCompactTime	Number	1,516,892,074
▼ HostUUIDs	Array	(1 item)
Item 0	String	502A32AD-3CAF-5585-BC09-053E285901C8
LocalizedDiskImageVolumeName	String	Time Machine Backups
BackupAlias	Data	<00000000 03c80002 00010444 61746100 00000000 00000000
PreferencesVersion	Number	4
LastConfigurationTraceDate	Date	Mar 24, 2018 at 9:28:36 PM
▼ Destinations	Array	(1 item)
▼ Item 0	Dictionary	(14 items)
BackupAlias	Data	<00000000 04100002 00010444 61746100 00000000 00000000
TimeCapsuleName	String	Delorean
ConsistencyScanDate	Date	Mar 9, 2018 at 6:38:01 AM
▼ DestinationUUIDs	Array	(2 items)
Item 0	String	AE9BC8DF-F13C-39EC-966D-1163A90B2138
Item 1	String	F567E2C0-64B2-31A4-BD86-674310C2B2E8
ReferenceLocalSnapshotDate	Date	Mar 11, 2018 at 1:30:19 PM
RESULT	Number	19
LastKnownEncryptionState	String	Encrypted
DateOfLatestWarning	Date	Mar 26, 2018 at 8:43:28 PM
DestinationID	String	552420F8-D5F2-4C94-8820-D2EFC0155A92
BytesUsed	Number	1,357,859,663,872
FirmwareCheckDate	Date	Mar 6, 2018 at 11:27:20 PM
▼ SnapshotDates	Array	(27 items)
Item 0	Date	Mar 9, 2018 at 6:38:01 AM
Item 1	Date	Mar 9, 2018 at 11:47:29 PM
Item 2	Date	Mar 10, 2018 at 12:48:00 PM
Item 25	Date	Mar 11, 2018 at 12:24:46 PM
Item 26	Date	Mar 11, 2018 at 1:33:22 PM
RootVolumeUUID	String	1D19162C-518C-3A34-A02C-2D428A4BC44E
BytesAvailable	Number	640,047,771,648
▼ SkipPaths	Array	(1 item)
Item 0	String	/Users/Shared/adi
AutoBackup	Boolean	NO

```
000  00 00 00 00 03 C8 00 02 00 01 04 44 61 74 61 00 00 00 00 00 00 00 00 00 00 00   ...........Data............
027  00 00 00 00 00 00 00 00 00 00 00 CA AE 2D 3E 48 2B 00 01 00 00 00 01 04 44 61 74   ............->H+.......Dat
054  61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   a.....................................
081  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
108  00 00 00 00 00 00 00 00 00 02 CA AE 2D 3E 00 00 00 00 00 00 00 00 FF FF FF FF 00   ............->.............
135  00 31 11 61 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 44 61 74 61 00 10 00 08   .1.as.............Data....
162  00 00 CA AE 65 7E 00 00 00 11 00 08 00 00 CA AE 65 7E 00 00 00 01 00 00 00 02 00   ....e~..........e~.........
189  09 44 61 74 61 3A 44 61 74 61 00 00 0E 00 0A 00 04 00 44 00 61 00 74 00 61 00 0F   .Data:Data........D.a.t.a..
216  00 0A 00 04 00 44 00 61 00 74 00 61 00 12 00 00 00 13 00 0D 2F 56 6F 6C 75 6D 65   .....D.a.t.a......../Volume
243  73 2F 44 61 74 61 00 00 09 02 85 02 85 61 66 70 6D 00 80 04 04 00 0D 00 26 00 46   s/Data.......afpm.......&.F
270  00 6A 00 AA 00 EA 01 2E 00 61 01 37 01 57 00 00 01 00 02 75 00 00 00 00 00 00 00   .j.......a.7.W.....u.......
297  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
324  08 44 65 6C 6F 72 65 61 6E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .Delorean.............................
351  00 00 00 00 00 00 00 00 00 04 44 61 74 61 00 00 00 00 00 00 00 00 00 00 00 00 00   ..........Data.....................
378  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
405  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 53 61 72 61 68 20 45   ..................Sarah E
432  64 77 61 72 64 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   dwards.....................
459  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
486  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
513  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
540  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
567  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
594  00 00 00 00 02 22 04 44 65 6C 6F 72 65 61 6E 2E 5F 61 66 70 6F 76 65 72 74 63 70   ....."·Delorean._afpovertcp
621  2E 5F 74 63 70 2E 6C 6F 63 61 6C 2E 08 02 0A 0B 0C 84 02 24 00 00 00 00 00 00 00   ._tcp.local.......$.......
648  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
675  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
702  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
729  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
756  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
783  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
810  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
837  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................................
864  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 53 61 72 61 68 20   .....................Sarah
891  45 64 77 61 72 64 73 00 00 00 18 00 3B 61 66 70 3A 2F 2F 53 61 72 61 68 25 32 30   Edwards.....;afp://Sarah%20
918  45 64 77 61 72 64 73 40 44 65 6C 6F 72 65 61 6E 2E 5F 61 66 70 6F 76 65 72 74 63   Edwards@Delorean._afpovertc
945  70 2E 5F 74 63 70 2E 6C 6F 63 61 6C 2E 2F 44 61 74 61 00 FF FF 00 00              p._tcp.local./Data.....
```

Time Machine: Backup Excluded Files /System/Library/CoreServices/backupd.bundle/Contents/Resources/StdExclusions.plist



Root	Dictionary	(4 items)
PathsExcluded	Array	(25 items)
Item 0	String	/.MobileBackups
Item 1	String	/MobileBackups.trash
Item 2	String	/.MobileBackups.trash
Item 3	String	/.Spotlight-V100
Item 4	String	/.TemporaryItems
Item 5	String	/.Trashes
Item 6	String	/.com.apple.backupd.mvlist.plist
Item 7	String	/.fseventsd
Item 8	String	/.hotfiles.btree
Item 9	String	/Backups.backupdb
Item 10	String	/Desktop DB
Item 11	String	/Desktop DF
Item 12	String	/Network/Servers
Item 13	String	/Library/Updates
Item 14	String	/Previous Systems
Item 15	String	/Users/Shared/SC Info
Item 16	String	/Users/Guest
Item 17	String	/dev
Item 18	String	/home
Item 19	String	/net
Item 20	String	/private/var/db/com.apple.backupd.backupVerification
Item 21	String	/private/var/db/efw_cache
Item 22	String	/private/var/db/Spotlight
Item 23	String	/private/var/db/Spotlight-V100
Item 24	String	/private/var/lib/postfix/greylist.db
ContentsExcluded	Array	(20 items)
FileContentsExcluded	Array	(4 items)
UserPathsExcluded	Array	(21 items)

The `StdExclusions.plist` property list file located in the `/System/Library/CoreServices/backupd.bundle/Contents/Resources/` directory contains the default Time Machine exclusions.

This list contains the directory file paths for items that should also be excluded in Time Machine backups.

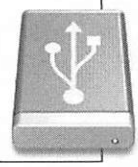
▼ Root	Dictionary	(4 items)
▼ PathsExcluded	Array	(25 items)
Item 0	String	/.MobileBackups
Item 1	String	/MobileBackups.trash
Item 2	String	/.MobileBackups.trash
Item 3	String	/.Spotlight-V100
Item 4	String	/.TemporaryItems
Item 5	String	/.Trashes
Item 6	String	/.com.apple.backupd.mvlist.plist
Item 7	String	/.fseventsd
Item 8	String	/.hotfiles.btree
Item 9	String	/Backups.backupdb
Item 10	String	/Desktop DB
Item 11	String	/Desktop DF
Item 12	String	/Network/Servers
Item 13	String	/Library/Updates
Item 14	String	/Previous Systems
Item 15	String	/Users/Shared/SC Info
Item 16	String	/Users/Guest
Item 17	String	/dev
Item 18	String	/home
Item 19	String	/net
Item 20	String	/private/var/db/com.apple.backupd.backupVerification
Item 21	String	/private/var/db/efw_cache
Item 22	String	/private/var/db/Spotlight
Item 23	String	/private/var/db/Spotlight-V100
Item 24	String	/private/var/lib/postfix/greylist.db
▶ ContentsExcluded	Array	(20 items)
▶ FileContentsExcluded	Array	(4 items)
▶ UserPathsExcluded	Array	(21 items)

Time Machine: Structure—External Backup Disk [1]

Volume Name	Available	Capacity	Mount Point	File System	BSD Name
TIMEMACHINE	113.34 GB	119.69 GB	/Volumes/TIMEMACHINE	Journaled HFS+	disk1s2
Macintosh HD	21.86 GB	239.71 GB	/	Journaled HFS+	disk0s2
ENCRYPTED TIMEMACHINE	53.48 GB	59.33 GB	/Volumes/ENCRYPTED TIMEMACHINE	Journaled HFS+	disk3

TIMEMACHINE:

Available: 113.34 GB (113,341,235,200 bytes)
Capacity: 119.69 GB (119,690,149,888 bytes)
Mount Point: /Volumes/TIMEMACHINE
File System: Journaled HFS+
Writable: Yes
Ignore Ownership: No
BSD Name: disk1s2
Volume UUID: 652BBBB8-CB3D-3A4D-97C8-5F7A6EF6441D
Physical Drive:
Media Name: WDC WD12 00VE-00KWT0 Media
Protocol: USB
Internal: No
Partition Map Type: GPT (GUID Partition Table)
S.M.A.R.T. Status: Not Supported



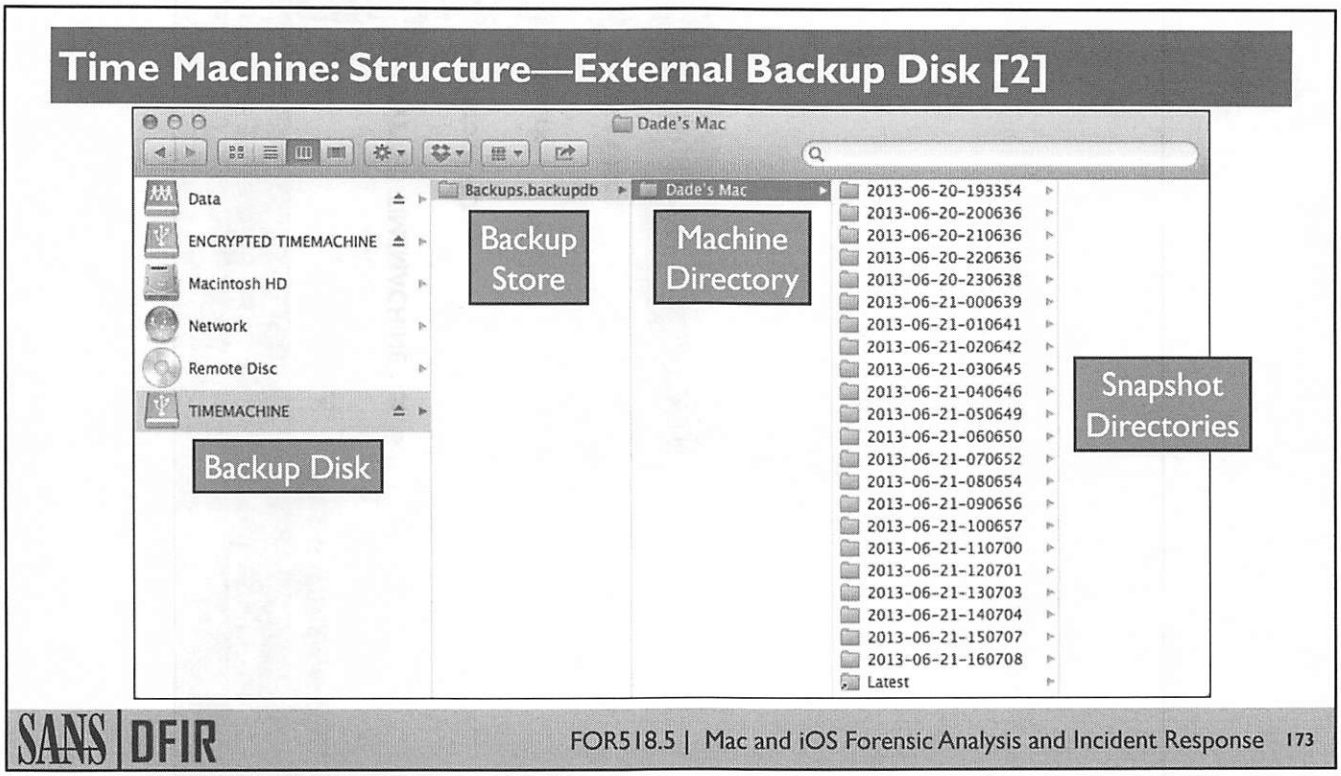
The TIMEMACHINE volume is shown in the screenshot above using the System Information application. This volume is using the HFS+ (Journaled) file system. All Time Machine volumes require the use of HFS+. It is not possible to have a Time Machine backup on FAT or NTFS formatted drives.

The screenshot also shows us the volume UUID, disk size, and connection protocol (USB).

Volume Name	Available	Capacity	Mount Point	File System	BSD Name
TIMEMACHINE	113.34 GB	119.69 GB	/Volumes/TIMEMACHINE	Journaled HFS+	disk1s2
Macintosh HD	21.86 GB	239.71 GB	/	Journaled HFS+	disk0s2
ENCRYPTED TIMEMACHINE	53.48 GB	59.33 GB	/Volumes/ENCRYPTED TIMEMACHINE	Journaled HFS+	disk3

TIMEMACHINE:

Available: 113.34 GB (113,341,235,200 bytes)
Capacity: 119.69 GB (119,690,149,888 bytes)
Mount Point: /Volumes/TIMEMACHINE
File System: Journaled HFS+
Writable: Yes
Ignore Ownership: No
BSD Name: disk1s2
Volume UUID: 652BBBB8-CB3D-3A4D-97C8-5F7A6EF6441D
Physical Drive:
    Media Name: WDC WD12 00VE-00KWT0 Media
    Protocol: USB
    Internal: No
    Partition Map Type: GPT (GUID Partition Table)
    S.M.A.R.T. Status: Not Supported

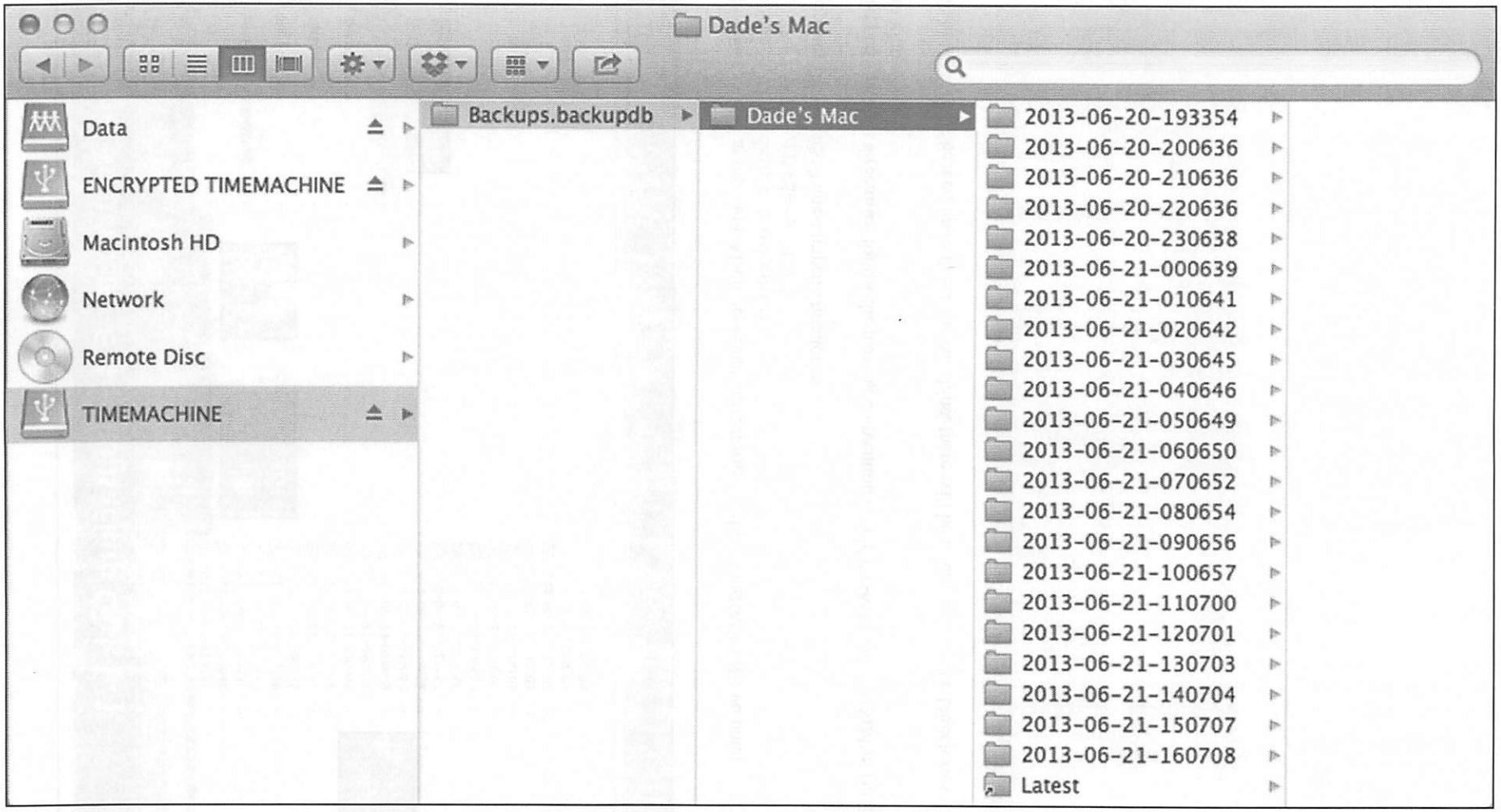


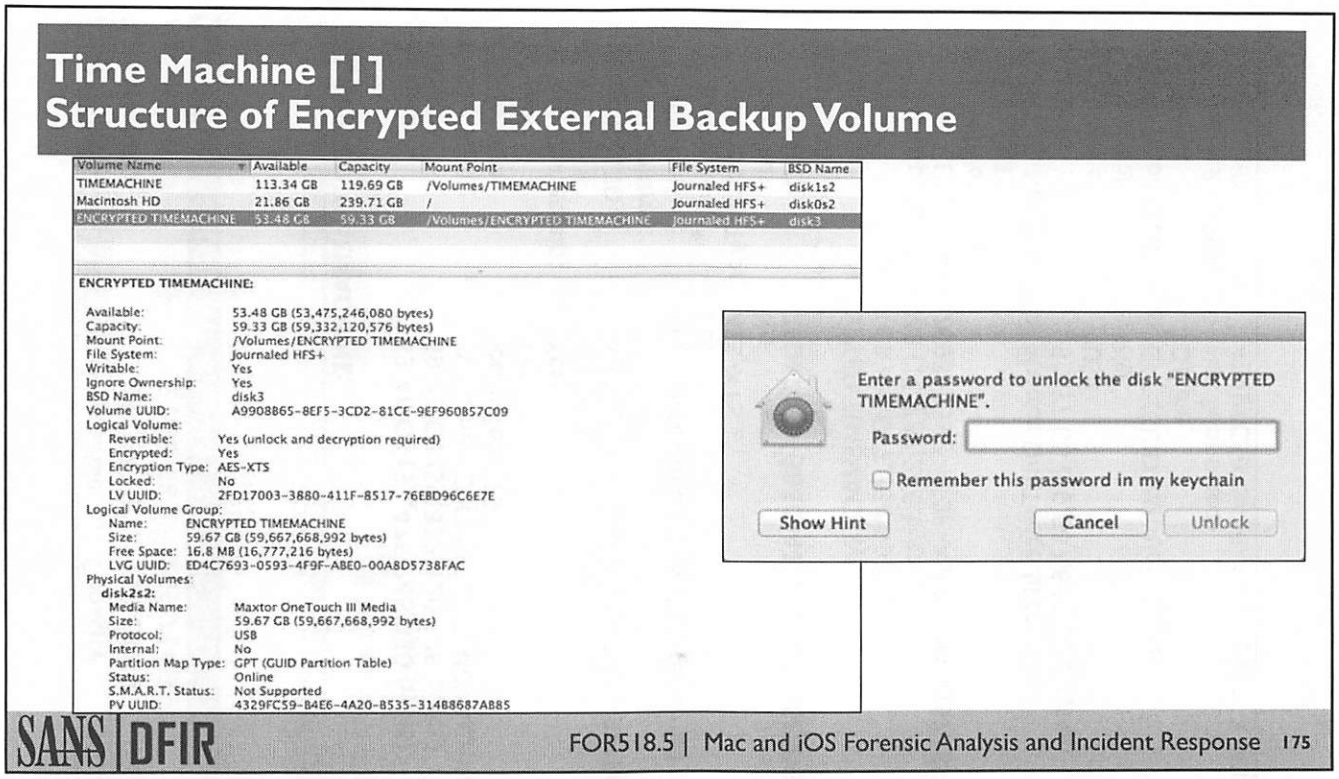
Using Finder, we can look at the Time Machine–related directories on the `TIMEMACHINE` volume:

- Backup Store: `Backups.backupdb/`
- Machine Directory: `Dade's Mac/`
- Snapshot Directories: Timestamped Directories

The timestamped snapshot directories follow the naming convention of YYYY-MM-DD-HHMMSS (in 24-hour local system time).

If more than one system is backed up to this Time Machine, there will be multiple Machine Directories.



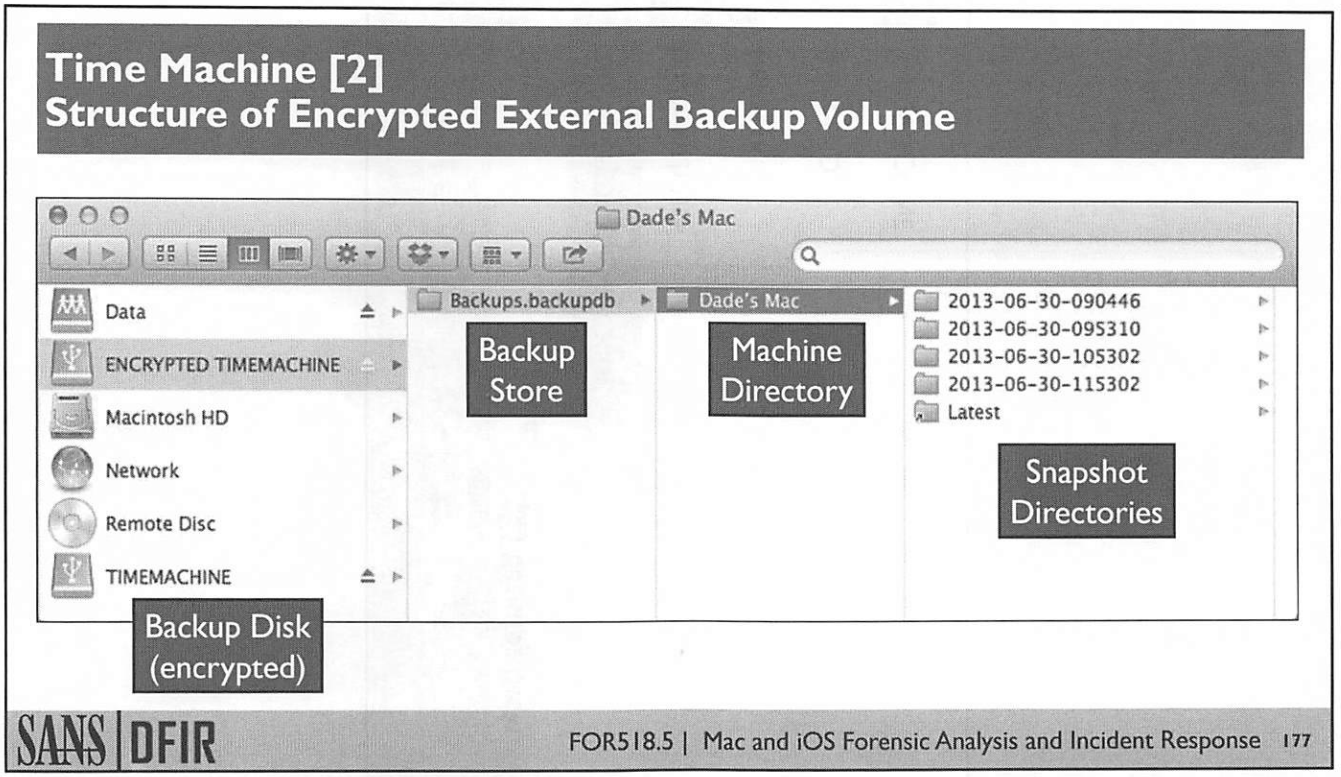


When an encrypted Time Machine is mounted (for example, from a forensic image), it will ask for a password, as shown in the screenshot above. This password allows access to mount the encrypted CoreStorage volume as shown in the larger screenshot above.

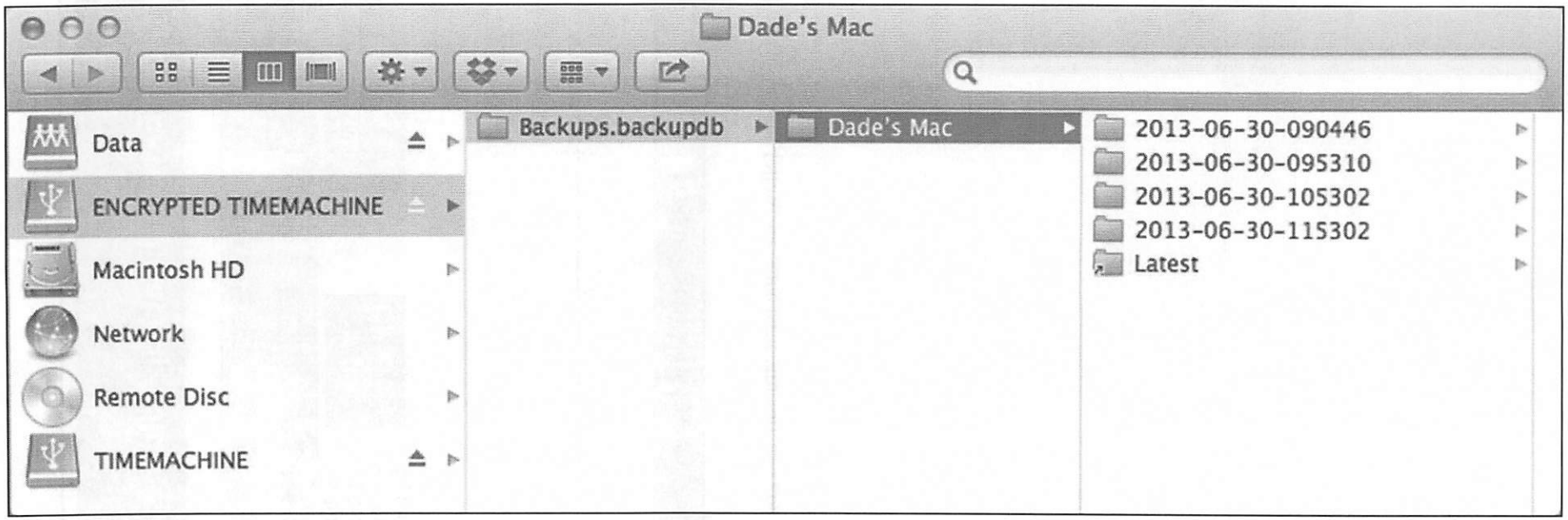
Volume Name	Available	Capacity	Mount Point	File System	BSD Name
TIMEMACHINE	113.34 GB	119.69 GB	/Volumes/TIMEMACHINE	Journaled HFS+	disk1s2
Macintosh HD	21.86 GB	239.71 GB	/	Journaled HFS+	disk0s2
ENCRYPTED TIMEMACHINE	53.48 GB	59.33 GB	/Volumes/ENCRYPTED TIMEMACHINE	Journaled HFS+	disk3

ENCRYPTED TIMEMACHINE:

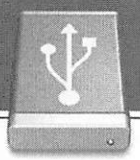
Available: 53.48 GB (53,475,246,080 bytes)
Capacity: 59.33 GB (59,332,120,576 bytes)
Mount Point: /Volumes/ENCRYPTED TIMEMACHINE
File System: Journaled HFS+
Writable: Yes
Ignore Ownership: Yes
BSD Name: disk3
Volume UUID: A9908B65-8EF5-3CD2-81CE-9EF960B57C09
Logical Volume:
  Revertible: Yes (unlock and decryption required)
  Encrypted: Yes
  Encryption Type: AES-XTS
  Locked: No
  LV UUID: 2FD17003-3880-411F-8517-76EBD96C6E7E
Logical Volume Group:
  Name: ENCRYPTED TIMEMACHINE
  Size: 59.67 GB (59,667,668,992 bytes)
  Free Space: 16.8 MB (16,777,216 bytes)
  LVG UUID: ED4C7693-0593-4F9F-ABE0-00A8D5738FAC
Physical Volumes:
  disk2s2:
    Media Name: Maxtor OneTouch III Media
    Size: 59.67 GB (59,667,668,992 bytes)
    Protocol: USB
    Internal: No
    Partition Map Type: GPT (GUID Partition Table)
    Status: Online
    S.M.A.R.T. Status: Not Supported
    PV UUID: 4329FC59-B4E6-4A20-B535-314B8687AB85



Once mounted, the encrypted backup volume follows the same basic structure as a non-encrypted external backup volume, including metadata.



Time Machine: External Disk Structure
Machine Directory Extended Attributes



```
nibble:Backups.backupdb sledwards$ xattr -xl Dade's\ Mac/
com.apple.backupd.BackupMachineAddress:
00000000  30 30 3A 30 63 3A 32 39 3A 39 39 3A 36 62 3A 38  |00:0c:29:99:6b:8|
00000010  65 00                                            |e.|
00000012
com.apple.backupd.HasRecoverySet:
00000000  59 45 53                                         |YES|
00000003
com.apple.backupd.HostUUID:
00000000  30 30 30 30 30 30 30 30 2D 30 30 30 30 2D 31 30  |00000000-0000-10|
00000010  30 30 2D 38 30 30 30 2D 30 30 30 43 32 39 39 39  |00-8000-000C2999|
00000020  36 42 38 45 00                                   |6B8E.|
00000025
com.apple.backupd.ModelID:
00000000  56 4D 77 61 72 65 37 2C 31                       |VMware7,1|
00000009
```


Using the extended attributes `xattr` command, we can view the attributes for the Machine Directory "`Dade's Mac`". The Machine Directory contains the following extended attributes:

- `com.apple.backupd.BackupMachineAddress`: Network MAC Address of the system
- `com.apple.backupd.HasRecoverySet`: Ability to boot from backup for recovery (look for the `Backups.backupdb/.RecoverySets` directory)
- `com.apple.backupd.HostUUID`: Hardware UUID
- `com.apple.backupd.ModelID`: Hardware Model Identifier

If the user selected "Encrypted Backups", you will find an additional extended attribute of "`com.apple.backupd.HasEncryptedRecoveryBits`" set to "`YES`".

Investigators can use this information to determine the type of system these backups are from and potentially find other systems to analyze.

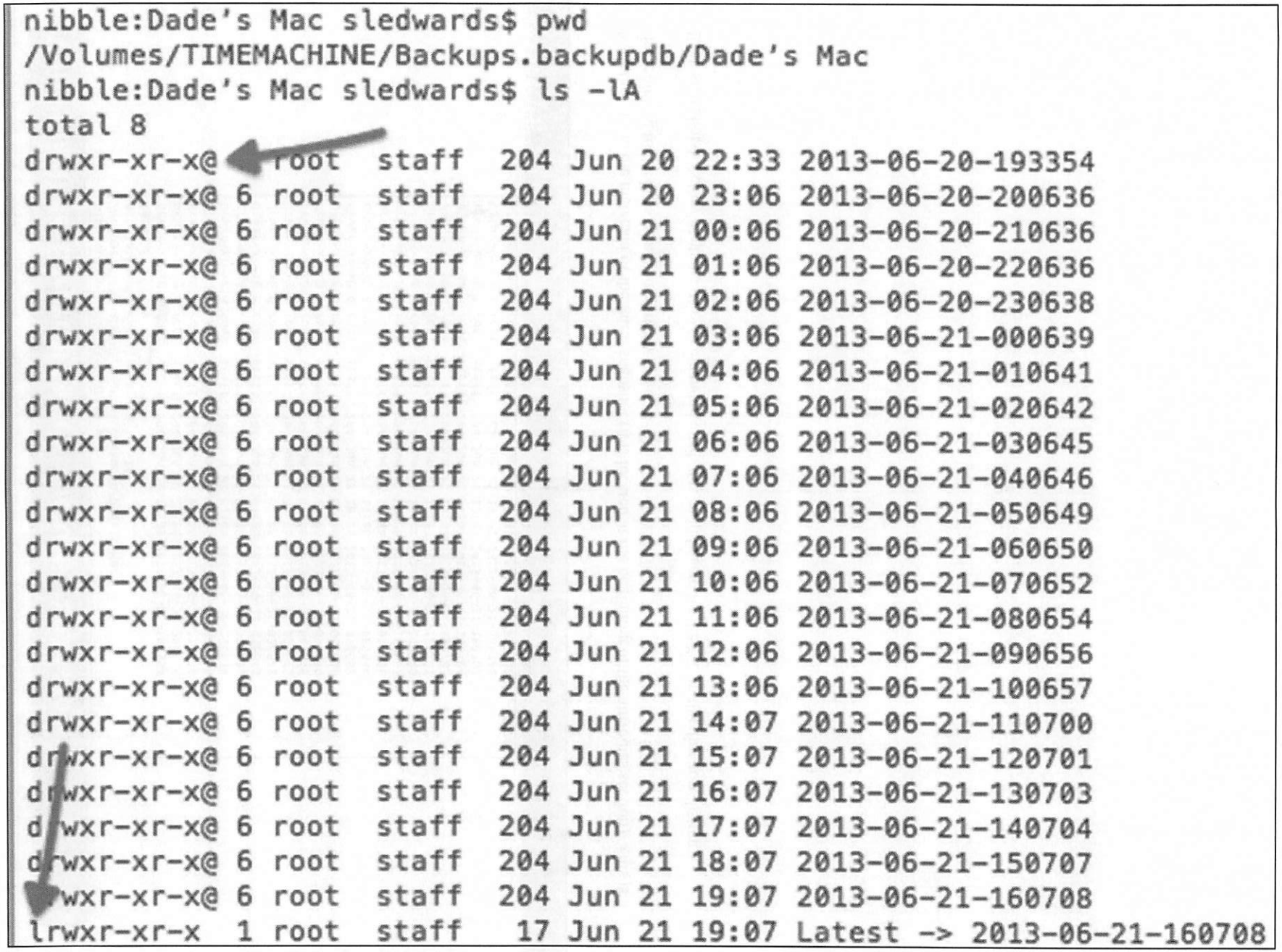
```
nibble:Backups.backupdb sledwards$ xattr -xl Dade's\ Mac/
com.apple.backupd.BackupMachineAddress:
00000000  30 30 3A 30 63 3A 32 39 3A 39 39 3A 36 62 3A 38  |00:0c:29:99:6b:8|
00000010  65 00                                            |e.|
00000012
com.apple.backupd.HasRecoverySet:
00000000  59 45 53                                         |YES|
00000003
com.apple.backupd.HostUUID:
00000000  30 30 30 30 30 30 30 30 2D 30 30 30 30 2D 31 30  |00000000-0000-10|
00000010  30 30 2D 38 30 30 30 2D 30 30 30 43 32 39 39 39  |00-8000-000C2999|
00000020  36 42 38 45 00                                   |6B8E.|
00000025
com.apple.backupd.ModelID:
00000000  56 4D 77 61 72 65 37 2C 31                       |VMware7,1|
00000009
```

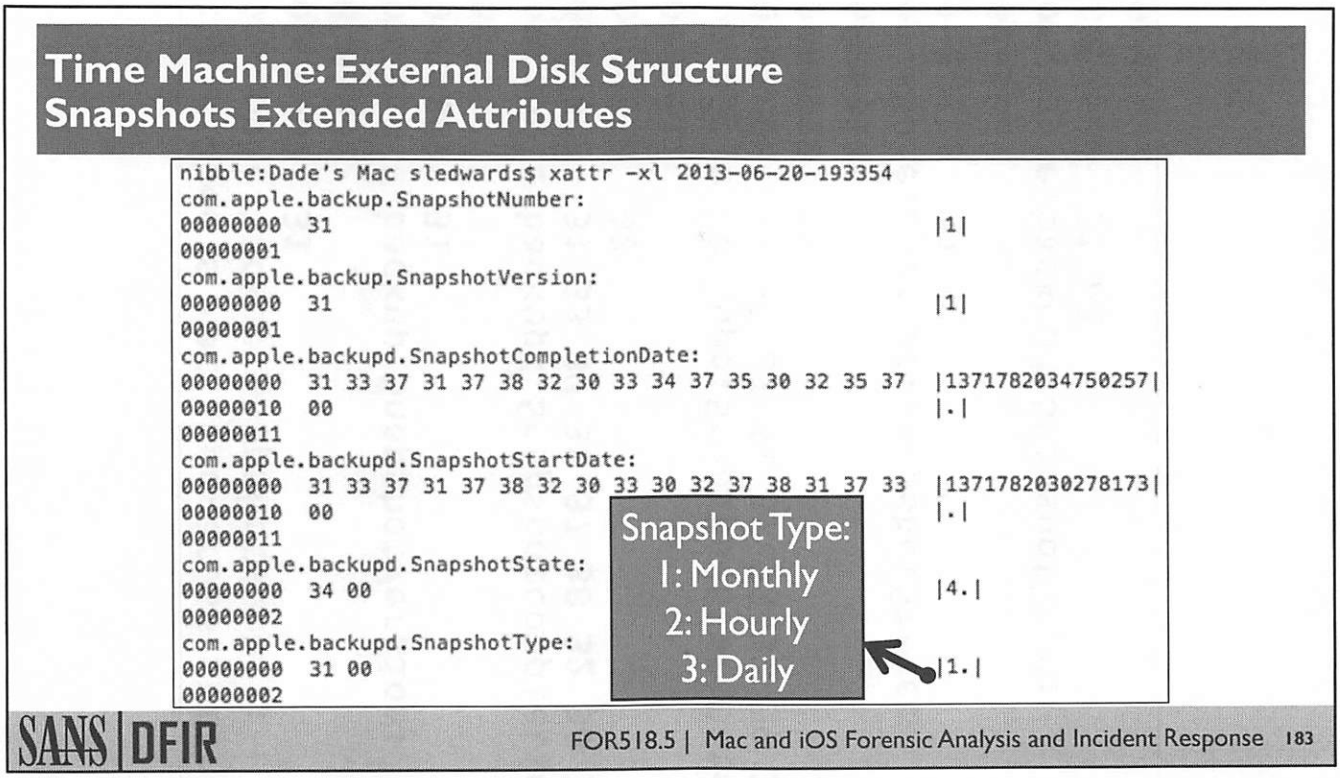
Time Machine: External Disk Structure—Snapshots

```
nibble:Dade's Mac sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac
nibble:Dade's Mac sledwards$ ls -lA
total 8
drwxr-xr-x@ 6 root  staff  204 Jun 20 22:33 2013-06-20-193354
drwxr-xr-x@ 6 root  staff  204 Jun 20 23:06 2013-06-20-200636
drwxr-xr-x@ 6 root  staff  204 Jun 21 00:06 2013-06-20-210636
drwxr-xr-x@ 6 root  staff  204 Jun 21 01:06 2013-06-20-220636
drwxr-xr-x@ 6 root  staff  204 Jun 21 02:06 2013-06-20-230638
drwxr-xr-x@ 6 root  staff  204 Jun 21 03:06 2013-06-21-000639
drwxr-xr-x@ 6 root  staff  204 Jun 21 04:06 2013-06-21-010641
drwxr-xr-x@ 6 root  staff  204 Jun 21 05:06 2013-06-21-020642
drwxr-xr-x@ 6 root  staff  204 Jun 21 06:06 2013-06-21-030645
drwxr-xr-x@ 6 root  staff  204 Jun 21 07:06 2013-06-21-040646
drwxr-xr-x@ 6 root  staff  204 Jun 21 08:06 2013-06-21-050649
drwxr-xr-x@ 6 root  staff  204 Jun 21 09:06 2013-06-21-060650
drwxr-xr-x@ 6 root  staff  204 Jun 21 10:06 2013-06-21-070652
drwxr-xr-x@ 6 root  staff  204 Jun 21 11:06 2013-06-21-080654
drwxr-xr-x@ 6 root  staff  204 Jun 21 12:06 2013-06-21-090656
drwxr-xr-x@ 6 root  staff  204 Jun 21 13:06 2013-06-21-100657
drwxr-xr-x@ 6 root  staff  204 Jun 21 14:07 2013-06-21-110700
drwxr-xr-x@ 6 root  staff  204 Jun 21 15:07 2013-06-21-120701
drwxr-xr-x@ 6 root  staff  204 Jun 21 16:07 2013-06-21-130703
drwxr-xr-x@ 6 root  staff  204 Jun 21 17:07 2013-06-21-140704
drwxr-xr-x@ 6 root  staff  204 Jun 21 18:07 2013-06-21-150707
drwxr-xr-x@ 6 root  staff  204 Jun 21 19:07 2013-06-21-160708
lrwxr-xr-x  1 root  staff   17 Jun 21 19:07 Latest -> 2013-06-21-160708
```


Note the "1" in the metadata of the latest snapshot (it is located at the beginning of the attributes section). This is a link pointing to the latest snapshot directory, `2013-06-21-160708`.

Each snapshot directory contains its own extended attributes as indicated by the "@" in the directory listing.





Using the extended attributes `xattr` command, we can view the attributes for the snapshot "`2013-06-20-193354`". This is the initial backup for the example system. This snapshot contains the following extended attributes:

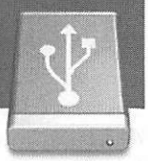
- `com.apple.backup.SnapshotNumber`: Incremental snapshot value (Note: It does not increment by single digits)
- `com.apple.backup.SnapshotVersion`: Appears to always be set to the value "1"
- `com.apple.backupd.SnapshotCompletionDate`: 16-digit Unix Epoch timestamp containing the backup completion date (input the first 10 digits into your favorite converter)
- `com.apple.backupd.SnapshotStartDate`: 16-digit Unix Epoch timestamp containing the start time of the backup
- `com.apple.backupd.SnapshotState`: Appears to always be set to the value "4" or `0x3400`
- `com.apple.backupd.SnapshotType`: Type of backup:
  - 1: Monthly
  - 2: Hourly
  - 3: Daily

10.9 systems added an additional attribute "`com.apple.backupd.SnapshotTotalBytesCopied`", which contains the number of bytes copied into this snapshot.

Analysts can use this information to determine when a particular snapshot was created and the type of snapshot it is.

```
nibble:Dade's Mac sledwards$ xattr -xl 2013-06-20-193354
com.apple.backup.SnapshotNumber:
00000000  31                                                |1|
00000001
com.apple.backup.SnapshotVersion:
00000000  31                                                |1|
00000001
com.apple.backupd.SnapshotCompletionDate:
00000000  31 33 37 31 37 38 32 30 33 34 37 35 30 32 35 37  |1371782034750257|
00000010  00                                                |.|
00000011
com.apple.backupd.SnapshotStartDate:
00000000  31 33 37 31 37 38 32 30 33 30 32 37 38 31 37 33  |1371782030278173|
00000010  00                                                |.|
00000011
com.apple.backupd.SnapshotState:
00000000  34 00                                             |4.|
00000002
com.apple.backupd.SnapshotType:
00000000  31 00                                             |1.|
00000002
```

Time Machine: External Disk Structure
Snapshot Contents



```
nibble:2013-06-20-193354 sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354
nibble:2013-06-20-193354 sledwards$ ls -la
total 16
drwxr-xr-x@  6 root  staff   204 Jun 20 22:33 .
drwxr-xr-x@ 25 root  staff   850 Jun 21 19:07 ..
-rw-------   1 root  staff  3668 Jun 20 22:33 .Backup.log
-rw-------   1 root  staff     0 Jun 20 22:33 .com.apple.TMCheckpoint
-rw-------   1 root  staff  2220 Jun 20 22:33 .exclusions.plist
drwxr-xr-x@ 23 root  wheel   782 Jun 20 22:33 Macintosh HD
```


The screenshot shows the contents of the snapshot `2013-06-20-193354`. The Snapshot Volume name is "`Macintosh HD`". This directory also contains the hidden files:

- `.Backup.log`: Contains the backup log
- `.com.apple.TMCheckpoint`: Unknown; size is always 0 bytes
- `.exclusions.plist`: Contains a list of the excluded files and directories for this snapshot

The `.Backup.log` file contains the backup log data for that specific snapshot. It will include items such as excluded directories, how much data was copied, and how long it took to create the snapshot. An example is shown on the following pages.

```
2013-06-20-19:05:56 - Starting backup

Previous snapshot:
        None

Will traverse "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-B3E3-
169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
=== Starting backup loop #1 ===
  Will use FirstBackupCopier

Running preflight for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-
B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
        Excluding /.Spotlight-V100: 50.6 MB (90 items)
        Excluding /.Trashes: Zero KB (1 items)
        Excluding /.fseventsd: 29 KB (8 items)
        Excluding /.hotfiles.btree: 66 KB (1 items)
        Excluding /Library/Updates: 1.03 GB (15 items)
        Excluding /private/var/db/Spotlight: Zero KB (2 items)
        Excluding /Volumes: 4 KB (4 items)
        Excluding /Network: Zero KB (1 items)
        Excluding /.vol: Zero KB (1 items)
        Excluding /cores: Zero KB (1 items)
        Excluding /private/tmp: Zero KB (7 items)
        Excluding /private/tftpboot: Zero KB (1 items)
        Excluding /private/var/folders: 504.4 MB (165 items)
        Excluding /private/var/run: 33 KB (15 items)
        Excluding /private/var/tmp: Zero KB (3 items)
        Excluding /private/var/vm: 67.1 MB (2 items)
        Excluding /private/var/db/dhcpclient: 4 KB (3 items)
        Excluding /Library/Caches: 12.7 MB (11 items)
        Excluding /Library/Logs: 4 KB (6 items)
        Excluding /System/Library/Caches: 22.1 MB (39 items)
        Excluding /private/var/log: 4.3 MB (92 items)
        Excluding /private/var/spool/cups: 8 KB (6 items)
        Excluding /private/var/spool/fax: Zero KB (1 items)
        Excluding /private/var/spool/uucp: Zero KB (1 items)
        Excluding /private/var/db/dyld: 523.6 MB (12 items)
        Should copy 326143 items (6.03 GB) representing 1471664 blocks of size
4096. 29166761 blocks available.
Preflight complete for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-
B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Time elapsed: 0.246 seconds

Processing preflight info
        Space needed for this backup: 7.23 GB (1766133 blocks of size 4096)
Finished processing preflight info
```

```
Copying items from "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-B3E3-
169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Finished copying items for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-
3335-B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Time elapsed: 27 minutes, 52.000 seconds
        Copied 325141 items (5.56 GB)
Gathering events since 28327.
Needs new backup due to change in /private/var/db/.dat0207.001
=== Starting backup loop #2 ===
  Will use IncrementalBackupCopier

Running preflight for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-
B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
        Calculating size of changes
        Should copy 14 items (Zero KB) representing 0 blocks of size 4096.
27710787 blocks available.
Preflight complete for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-
B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Time elapsed: 0.245 seconds

Processing preflight info
        Space needed for this backup: 402.5 MB (98257 blocks of size 4096)
        Preserving last snapshot /Volumes/Untitled/Backups.backupdb/Dade's
Mac/2013-06-20-190556.inProgress/1F682AE6-31FF-42A5-9430-4D47C1BBB689
Finished processing preflight info

Copying items from "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-3335-B3E3-
169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Finished copying items for "Macintosh HD" (mount: '/' fsUUID: 0A81F3B1-51D9-
3335-B3E3-169C3640360D eventDBUUID: 23E6B5CA-AFD4-4150-B08E-7186D734FC35)
Time elapsed: 4.122 seconds
        Copied 233 items (33 bytes)
Gathering events since 4782116.

Backup complete.
Total time elapsed: 27 minutes, 59.000 seconds
```

Time Machine: External Disk Structure
Snapshot Volume Extended Attributes

```
nibble:2013-06-20-193354 sledwards$ xattr -xl Macintosh\ HD/
com.apple.FinderInfo:
00000000  00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000020
com.apple.backupd.SnapshotVolumeFSEventStoreUUID:
00000000  32 33 45 36 42 35 43 41 2D 41 46 44 34 2D 34 31  |23E6B5CA-AFD4-41|
00000010  35 30 2D 42 30 38 45 2D 37 31 38 36 44 37 33 34  |50-B08E-7186D734|
00000020  46 43 33 35 00                                    |FC35.|
00000025
com.apple.backupd.SnapshotVolumeLastFSEventID:
00000000  34 37 38 32 31 31 36 00                           |4782116.|
00000008
com.apple.backupd.SnapshotVolumeUUID:
00000000  30 41 38 31 46 33 42 31 2D 35 31 44 39 2D 33 33  |0A81F3B1-51D9-33|
00000010  33 35 2D 42 33 45 33 2D 31 36 39 43 33 36 34 30  |35-B3E3-169C3640|
00000020  33 36 30 44 00                                    |360D.|
00000025
com.apple.backupd.VolumeBytesUsed:
00000000  38 33 30 39 32 38 34 38 36 34                    |8309284864|
0000000a
com.apple.backupd.VolumeIsCaseSensitive:
00000000  30                                                |0|
00000001
com.apple.metadata:_kTimeMachineNewestSnapshot:
00000000  62 70 6C 69 73 74 30 30 33 41 B7 73 F3 12 00 00  |bplist003A.s....|
00000010  00 08 00 00 00 00 00 00 00 01 01 00 00 00 00 00  |................|
00000020  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000030  00 11                                             |..|
00000032
com.apple.metadata:_kTimeMachineOldestSnapshot:
00000000  62 70 6C 69 73 74 30 30 33 41 B7 73 F3 0E 00 00  |bplist003A.s....|
00000010  00 08 00 00 00 00 00 00 00 01 01 00 00 00 00 00  |................|
00000020  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000030  00 11                                             |..|
00000032
```


Using the extended attributes `xattr` command, we can view the attributes for the Snapshot Volume "`Macintosh HD`".

- `com.apple.FinderInfo`: Finder Attributes
- `com.apple.backupd.SnapshotVolumeFSEventStoreUUID`: UUID of the Events Store
- `com.apple.backupd.SnapshotVolumeLastFSEventID`: Last Event Store ID
- `com.apple.backupd.SnapshotVolumeUUID`: UUID of this Snapshot Volume
- `com.apple.backupd.VolumeBytesUsed`: Snapshot Volume size (if hard links are taken into account)
- `com.apple.backupd.VolumeIsCaseSensitive`: Case Sensitivity of Volume (0 = Not Case Sensitive)
- `com.apple.metadata:_kTimeMachineNewestSnapshot`: Binary plist containing the timestamp associated with the snapshot
- `com.apple.metadata:_kTimeMachineOldestSnapshot`: Binary plist containing the timestamp associated with the snapshot

This information can be used to determine disk usage between snapshots. Knowing if a large amount of files were deleted or added to the backup may be important to an investigator.

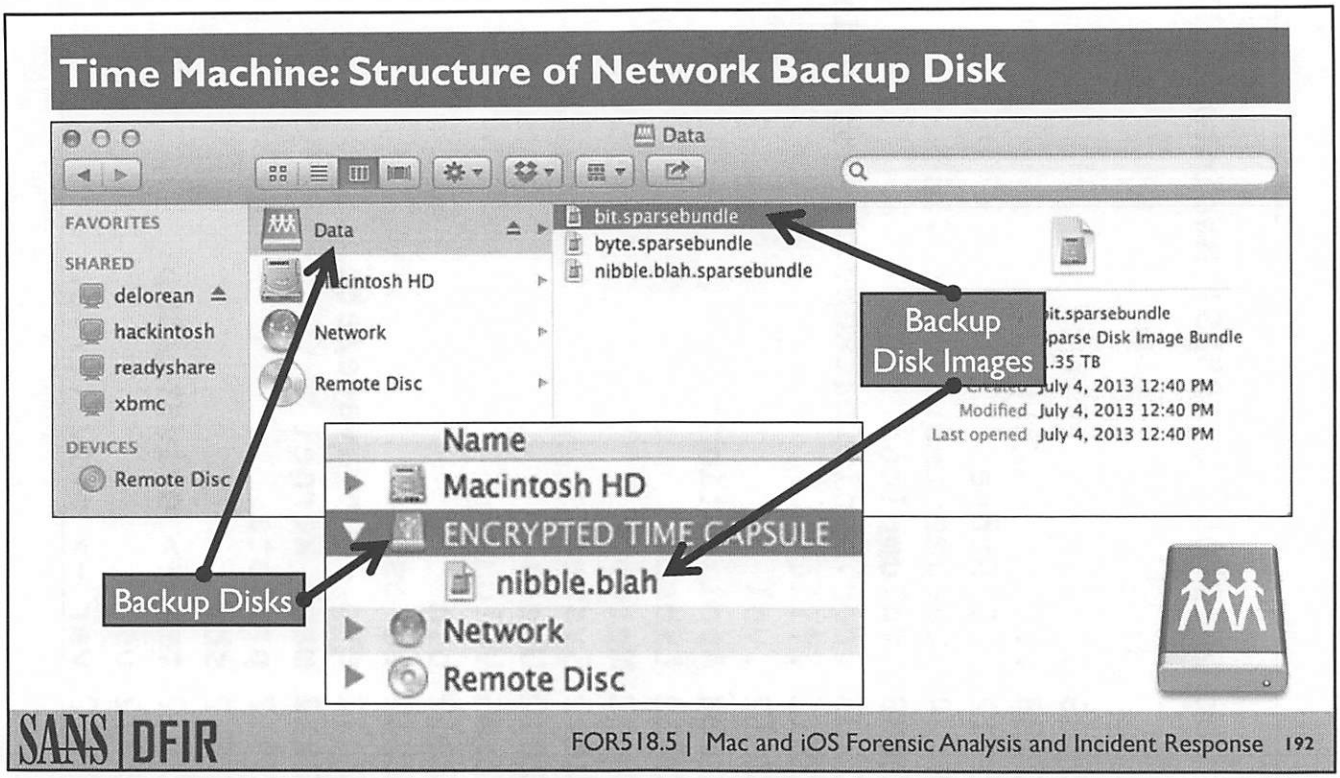
```
nibble:2013-06-20-193354 sledwards$ xattr -xl Macintosh\ HD/
com.apple.FinderInfo:
00000000  00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000020
com.apple.backupd.SnapshotVolumeFSEventStoreUUID:
00000000  32 33 45 36 42 35 43 41 2D 41 46 44 34 2D 34 31  |23E6B5CA-AFD4-41|
00000010  35 30 2D 42 30 38 45 2D 37 31 38 36 44 37 33 34  |50-B08E-7186D734|
00000020  46 43 33 35 00                                    |FC35.|
00000025
com.apple.backupd.SnapshotVolumeLastFSEventID:
00000000  34 37 38 32 31 31 36 00                          |4782116.|
00000008
com.apple.backupd.SnapshotVolumeUUID:
00000000  30 41 38 31 46 33 42 31 2D 35 31 44 39 2D 33 33  |0A81F3B1-51D9-33|
00000010  33 35 2D 42 33 45 33 2D 31 36 39 43 33 36 34 30  |35-B3E3-169C3640|
00000020  33 36 30 44 00                                    |360D.|
00000025
com.apple.backupd.VolumeBytesUsed:
00000000  38 33 30 39 32 38 34 38 36 34                    |8309284864|
0000000a
com.apple.backupd.VolumeIsCaseSensitive:
00000000  30                                                |0|
00000001
com.apple.metadata:_kTimeMachineNewestSnapshot:
00000000  62 70 6C 69 73 74 30 30 33 41 B7 73 F3 12 00 00  |bplist003A.s....|
00000010  00 08 00 00 00 00 00 00 01 01 00 00 00 00 00 00  |................|
00000020  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000030  00 11                                            |..|
00000032
com.apple.metadata:_kTimeMachineOldestSnapshot:
00000000  62 70 6C 69 73 74 30 30 33 41 B7 73 F3 0E 00 00  |bplist003A.s....|
00000010  00 08 00 00 00 00 00 00 01 01 00 00 00 00 00 00  |................|
00000020  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000030  00 11                                            |..|
00000032
```

Time Machine: External Disk Structure
Snapshot Volume Contents

```
nibble:Macintosh HD sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354/Macintosh HD
nibble:Macintosh HD sledwards$ ls -la
total 16304
drwxr-xr-x@ 23 root  wheel      782 Jun 20 22:33 .
drwxr-xr-x@  6 root  staff      204 Jun 20 22:33 ..
-rw-rw-r--@ 22 root  admin     6148 Sep 23  2012 .DS_Store
d--x--x--x   7 root  wheel      238 Apr 13 20:24 .DocumentRevisions-V100
-rw-r--r--@ 22 1000  staff   130756 Aug 27  2008 .VolumeIcon.icns
-rw-r--r--@  1 root  wheel      181 Jun 20 22:33 .com.apple.backupd.mvlist.plist
----------+ 22 root  admin        0 Jun 20  2012 .file
drwxr-xr-x@  2 root  wheel       68 Jun 20  2012 .vol
drwxrwxr-x@ 34 root  admin     1156 May 18 21:08 Applications
drwxrwxr-t@ 57 root  admin     1938 Sep 23  2012 Library
drwxr-xr-x@  2 root  wheel       68 Jun 20  2012 Network
drwxr-xr-x@  4 root  wheel      136 Sep 23  2012 System
drwxr-xr-x@  5 root  admin      170 Sep 23  2012 Users
drwxrwxrwt@  2 root  admin       68 Jun 20 22:05 Volumes
drwxr-xr-x@ 39 root  wheel     1326 Sep 23  2012 bin
drwxrwxr-t@  2 root  admin       68 Jun 20  2012 cores
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 etc -> private/etc
-rw-r--r--@ 22 root  wheel  8191712 Jun 25  2012 mach_kernel
drwxr-xr-x@  6 root  wheel      204 Sep 23  2012 private
drwxr-xr-x@ 63 root  wheel     2142 Sep 23  2012 sbin
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 tmp -> private/tmp
drwxr-xr-x@ 10 root  wheel      340 Sep 23  2012 usr
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 var -> private/var
```


Looking into the Snapshot Volume "`Macintosh HD`" we can see a normal OS X file system. While it may look complete, behind the scenes there are many hard links to various files. This allows snapshots to link to all the data files while not having to have multiple copies of redundant data. These links point to the file in the snapshot that contains the original version of the file.

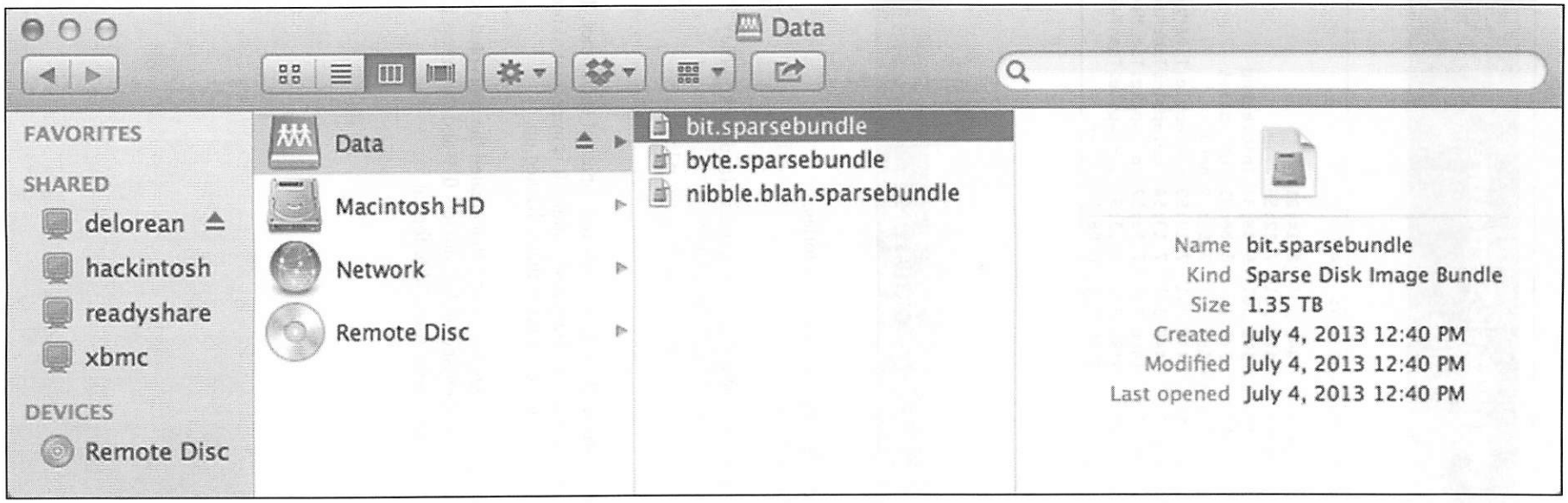
```
nibble:Macintosh HD sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354/Macintosh HD
nibble:Macintosh HD sledwards$ ls -la
total 16304
drwxr-xr-x@ 23 root  wheel      782 Jun 20 22:33 .
drwxr-xr-x@  6 root  staff      204 Jun 20 22:33 ..
-rw-rw-r--@ 22 root  admin     6148 Sep 23  2012 .DS_Store
d--x--x--x   7 root  wheel      238 Apr 13 20:24 .DocumentRevisions-V100
-rw-r--r--@ 22 1000  staff   130756 Aug 27  2008 .VolumeIcon.icns
-rw-r--r--@  1 root  wheel      181 Jun 20 22:33 .com.apple.backupd.mvlist.plist
----------+ 22 root  admin        0 Jun 20  2012 .file
drwxr-xr-x@  2 root  wheel       68 Jun 20  2012 .vol
drwxrwxr-x@ 34 root  admin     1156 May 18 21:08 Applications
drwxrwxr-t@ 57 root  admin     1938 Sep 23  2012 Library
drwxr-xr-x@  2 root  wheel       68 Jun 20  2012 Network
drwxr-xr-x@  4 root  wheel      136 Sep 23  2012 System
drwxr-xr-x@  5 root  admin      170 Sep 23  2012 Users
drwxrwxrwt@  2 root  admin       68 Jun 20 22:05 Volumes
drwxr-xr-x@ 39 root  wheel     1326 Sep 23  2012 bin
drwxrwxr-t@  2 root  admin       68 Jun 20  2012 cores
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 etc -> private/etc
-rw-r--r--@ 22 root  wheel  8191712 Jun 25  2012 mach_kernel
drwxr-xr-x@  6 root  wheel      204 Sep 23  2012 private
drwxr-xr-x@ 63 root  wheel     2142 Sep 23  2012 sbin
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 tmp -> private/tmp
drwxr-xr-x@ 10 root  wheel      340 Sep 23  2012 usr
lrwxr-xr-x@  1 root  wheel       11 Sep 23  2012 var -> private/var
```



Both non-encrypted and encrypted network backup disks use the sparse bundle file format.

The top screenshot shows an example from an unencrypted disk, while the inset screenshot shows an example from an encrypted Time Capsule volume.

Note: While the icon in the lower screenshot shows a USB drive, this is an external hard drive that was connected to a Time Capsule as a secondary network hard drive.



Time Machine: Network Disk Structure
Network Sparse Bundle Files

```
nibble:nibble.blah.sparsebundle sledwards$ ls -la
total 296
drwx------@    10 sledwards  staff     340 Jul  7 14:21 .
drwxrwxr-x     10 sledwards  staff     408 Jul  7 12:56 ..
-rw-r--r--      1 sledwards  staff     500 Jul  6 08:31 Info.bckup
-rw-r--r--      1 sledwards  staff     500 Jul  6 08:31 Info.plist
drwxr-xr-x  23544 sledwards  staff  800496 Jul  7 13:52 bands
-rw-r--r--      1 sledwards  staff     445 Jul  7 13:50 com.apple.TimeMachine.MachineID.bckup
-rw-r--r--      1 sledwards  staff     445 Jul  7 13:50 com.apple.TimeMachine.MachineID.plist
-rw-rw-rw-      1 sledwards  staff    1460 Jul  7 13:52 com.apple.TimeMachine.Results.plist
-rw-rw-rw-      1 sledwards  staff    4191 Jul  7 13:52 com.apple.TimeMachine.SnapshotHistory.plist
-rwx------      1 sledwards  staff  122368 Jul  6 08:31 token
```

Each sparse bundle contains the following files and directories:

- `Info.plist` (and backup file)
- Bands directory
- `com.apple.TimeMachine.MachineID.plist` (and backup file)
- `com.apple.TimeMachine.Results.plist`
- `com.apple.TimeMachine.SnapshotHistory.plist`
- Token file

`Info.plist` and `Info.bckup` are XML property lists and copies of each other. These property lists show the sparse bundle band size and allocated space of the sparse bundle. It is important to note that the "allocated space" in the `size` key may be much larger than the actual disk that is backed up.

The `bands` directory contains the sparse bundle `bands` that created the encrypted volume. Each `band` file has a unique sequential alphanumeric name comprised of the numbers 0–9 and the letters a–f. The `band` files contain bits and pieces of the backed-up data in a non-human-readable format.

```
nibble:nibble.blah.sparsebundle sledwards$ ls -la
total 296
drwx------@    10 sledwards  staff     340 Jul  7 14:21 .
drwxrwxr-x     10 sledwards  staff     408 Jul  7 12:56 ..
-rw-r--r--      1 sledwards  staff     500 Jul  6 08:31 Info.bckup
-rw-r--r--      1 sledwards  staff     500 Jul  6 08:31 Info.plist
drwxr-xr-x  23544 sledwards  staff  800496 Jul  7 13:52 bands
-rw-r--r--      1 sledwards  staff     445 Jul  7 13:50 com.apple.TimeMachine.MachineID.bckup
-rw-r--r--      1 sledwards  staff     445 Jul  7 13:50 com.apple.TimeMachine.MachineID.plist
-rw-rw-rw-      1 sledwards  staff    1460 Jul  7 13:52 com.apple.TimeMachine.Results.plist
-rw-rw-rw-      1 sledwards  staff    4191 Jul  7 13:52 com.apple.TimeMachine.SnapshotHistory.plist
-rwx------      1 sledwards  staff  122368 Jul  6 08:31 token
```

Time Machine: Network Disk Structure
com.apple.TimeMachine.MachineID.plist (.bckup)

▼ Root	Dictionary	(4 items)
VerificationDate	Date	Jul 6, 2013 5:33:12 AM
VerificationExtendedSkip	Boolean	NO
VerificationState	Number	1
com.apple.backupd.HostUUID	String	40A90B07-FC53-52C8-A774-6F1A5E659E9C

The property list files `com.apple.TimeMachine.MachineID.plist` and `com.apple.TimeMachine.MachineID.bckup` each contain the same information. These files contain the Host UUID and Time Machine verification data.

Time Machine: Network Disk Structure
com.apple.TimeMachine.Results.plist

Root	Dictionary	(18 items)
BACKUP_COMPLETED_DATE	Date	Jul 7, 2013 10:52:52 AM
BlockSize	Number	4,096
BlocksAvailable	Number	439,873,230
BlocksToCopy	Number	28
BlocksUsed	Number	48,337,460
BytesAvailable	Number	1,801,720,750,080
BytesToCopy	Number	116,091
BytesUsed	Number	197,990,236,160
ClientID	String	com.apple.backupd
PaddedBytesRequired	Number	1,068,996,173
Percent	Number	1
Progress	Dictionary	(6 items)
TimeRemaining	Number	-1
_raw_totalBytes	Number	17,932,597
bytes	Number	279,542,457
files	Number	979
totalBytes	Number	279,542,457
totalFiles	Number	979
RESULT	Number	0
Running	Boolean	YES
SnapshotCount	Number	14
_raw_Percent	Number	1
com.apple.backupd.SnapshotTotalBytesCopied	Number	279,542,457
kCSBackupdOldestCompleteSnapshotDate	Date	Jul 6, 2013 12:40:22 PM

The `com.apple.TimeMachine.Results.plist` contains data about the backup process, including:

- Backup completion date
- Disk block size and availability
- Disk byte size and availability
- Backup progress
- Snapshot data

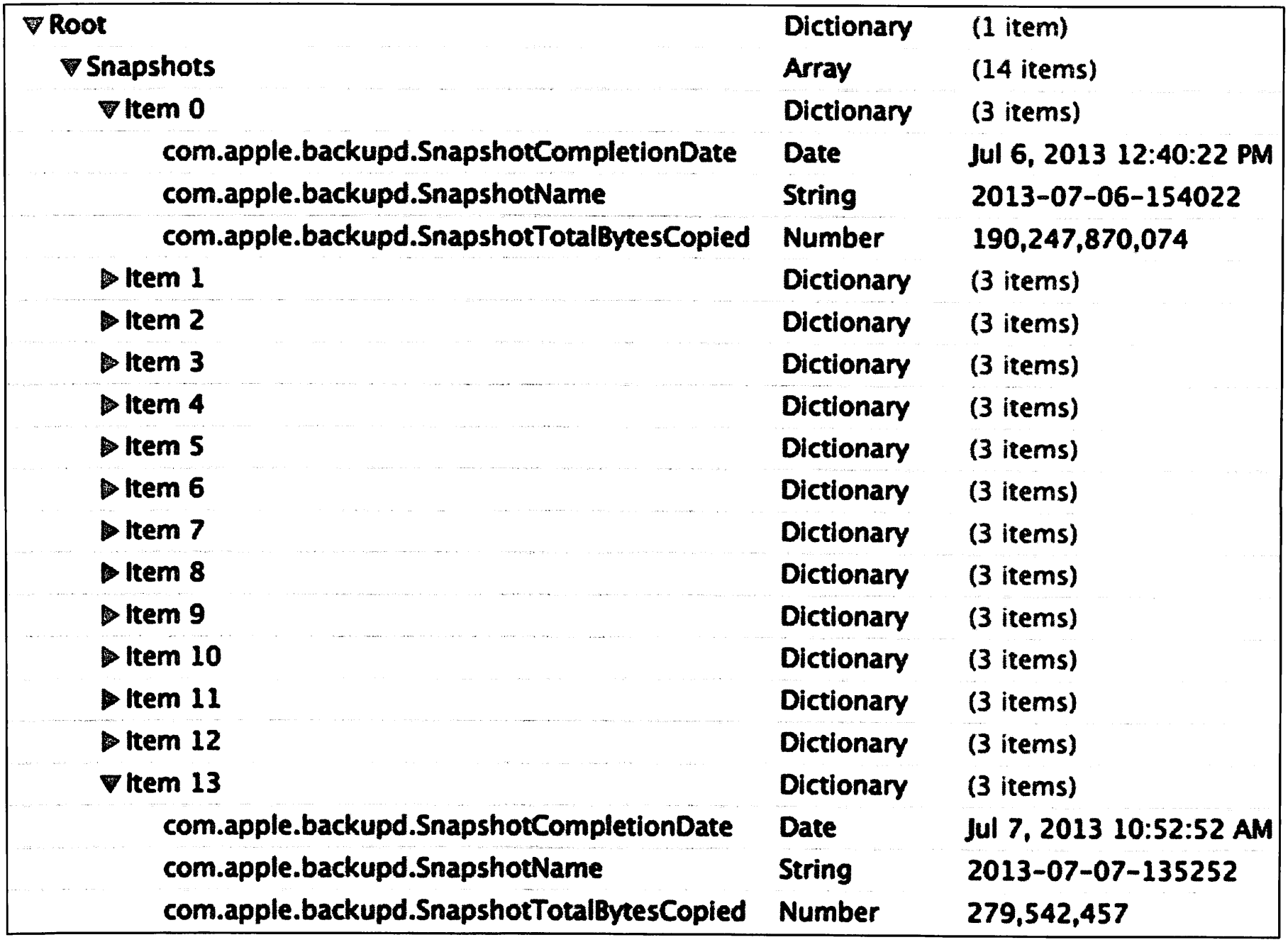
▼ Root	Dictionary	(18 items)
BACKUP_COMPLETED_DATE	Date	Jul 7, 2013 10:52:52 AM
BlockSize	Number	4,096
BlocksAvailable	Number	439,873,230
BlocksToCopy	Number	28
BlocksUsed	Number	48,337,460
BytesAvailable	Number	1,801,720,750,080
BytesToCopy	Number	116,091
BytesUsed	Number	197,990,236,160
ClientID	String	com.apple.backupd
PaddedBytesRequired	Number	1,068,996,173
Percent	Number	1
▼ Progress	Dictionary	(6 items)
TimeRemaining	Number	-1
_raw_totalBytes	Number	17,932,597
bytes	Number	279,542,457
files	Number	979
totalBytes	Number	279,542,457
totalFiles	Number	979
RESULT	Number	0
Running	Boolean	YES
SnapshotCount	Number	14
_raw_Percent	Number	1
com.apple.backupd.SnapshotTotalBytesCopied	Number	279,542,457
kCSBackupdOldestCompleteSnapshotDate	Date	Jul 6, 2013 12:40:22 PM

Time Machine: Network Disk Structure
com.apple.TimeMachine.SnapshotHistory.plist

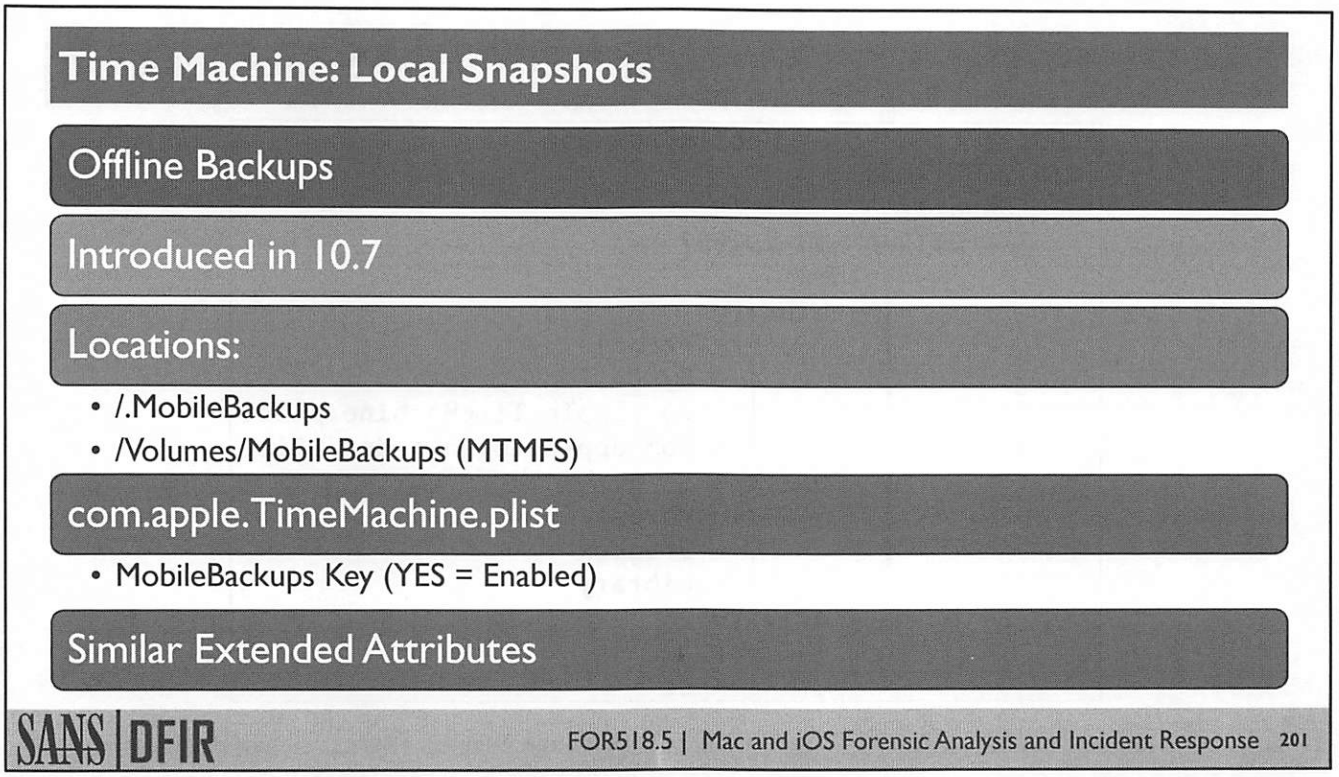
▼ Root	Dictionary	(1 item)
▼ Snapshots	Array	(14 items)
▼ Item 0	Dictionary	(3 items)
com.apple.backupd.SnapshotCompletionDate	Date	Jul 6, 2013 12:40:22 PM
com.apple.backupd.SnapshotName	String	2013-07-06-154022
com.apple.backupd.SnapshotTotalBytesCopied	Number	190,247,870,074
▶ Item 1	Dictionary	(3 items)
▶ Item 2	Dictionary	(3 items)
▶ Item 3	Dictionary	(3 items)
▶ Item 4	Dictionary	(3 items)
▶ Item 5	Dictionary	(3 items)
▶ Item 6	Dictionary	(3 items)
▶ Item 7	Dictionary	(3 items)
▶ Item 8	Dictionary	(3 items)
▶ Item 9	Dictionary	(3 items)
▶ Item 10	Dictionary	(3 items)
▶ Item 11	Dictionary	(3 items)
▶ Item 12	Dictionary	(3 items)
▼ Item 13	Dictionary	(3 items)
com.apple.backupd.SnapshotCompletionDate	Date	Jul 7, 2013 10:52:52 AM
com.apple.backupd.SnapshotName	String	2013-07-07-135252
com.apple.backupd.SnapshotTotalBytesCopied	Number	279,542,457

The `com.apple.TimeMachine.SnapshotHistory.plist` property list contains data about the stored Time Machine snapshots. This backup contains 14 snapshots, as shown in the screenshot above. Each snapshot `Item` has three keys associated with it:

- `com.apple.backupd.SnapshotCompletionDate`: Snapshot completion timestamp
- `com.apple.backupd.SnapshotName`: Snapshot name when sparse bundle is mounted
- `com.apple.backupd.SnapshotTotalBytesCopied`: Bytes copied by snapshot



▼ Root	Dictionary	(1 item)
▼ Snapshots	Array	(14 items)
▼ Item 0	Dictionary	(3 items)
com.apple.backupd.SnapshotCompletionDate	Date	Jul 6, 2013 12:40:22 PM
com.apple.backupd.SnapshotName	String	2013-07-06-154022
com.apple.backupd.SnapshotTotalBytesCopied	Number	190,247,870,074
▶ Item 1	Dictionary	(3 items)
▶ Item 2	Dictionary	(3 items)
▶ Item 3	Dictionary	(3 items)
▶ Item 4	Dictionary	(3 items)
▶ Item 5	Dictionary	(3 items)
▶ Item 6	Dictionary	(3 items)
▶ Item 7	Dictionary	(3 items)
▶ Item 8	Dictionary	(3 items)
▶ Item 9	Dictionary	(3 items)
▶ Item 10	Dictionary	(3 items)
▶ Item 11	Dictionary	(3 items)
▶ Item 12	Dictionary	(3 items)
▼ Item 13	Dictionary	(3 items)
com.apple.backupd.SnapshotCompletionDate	Date	Jul 7, 2013 10:52:52 AM
com.apple.backupd.SnapshotName	String	2013-07-07-135252
com.apple.backupd.SnapshotTotalBytesCopied	Number	279,542,457



When the Time Machine volume is otherwise inaccessible due to lack of network access or the Time Machine volume is not mounted, the system (if configured) will create local snapshots of the system itself. These are similar in format to normal Time Machine backups; however, they are saved to the local hard drive (if space permits).

This functionality was introduced in 10.7. This feature may be enabled on laptops and is disabled on desktop systems by default.

You may find these artifacts in two directories, depending on whether or not you are looking at a forensic image or a live system.

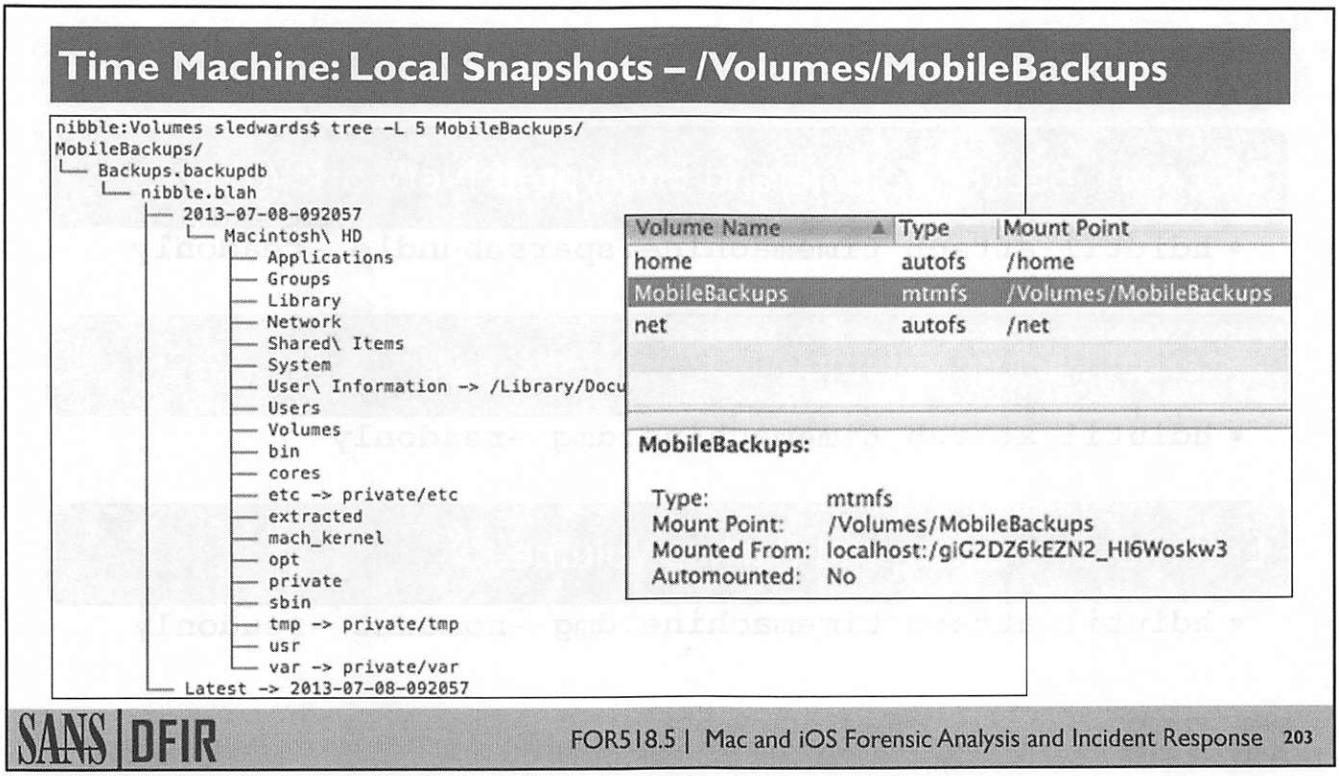
- `/.MobileBackups`: This directory will exist in the root of the system, for example, in a forensic image of a system.
- `/Volumes/MobileBackups`: On a live system, the `MobileBackups` volume may be mounted. This is a Mobile Time Machine File System (MTMFS) volume.

References:
`mtmd` Man Page
`mtmfs` Man Page
`https://support.apple.com/en-us/HT204015`

Time Machine: Local Snapshots – /.MobileBackups

```
sh-3.2# tree -L 6 .MobileBackups/
.MobileBackups/
└── Computer
    └── 2013-07-08-092057
        └── Volume
            ├── Library
            │   └── Preferences
            │       ├── SystemConfiguration
            │       ├── com.apple.TimeMachine.plist
            │       └── com.apple.loginwindow.plist
            ├── Users
            │   └── sledwards
            │       ├── Dropbox
            │       └── Library
            └── private
                └── var
                    └── db
```

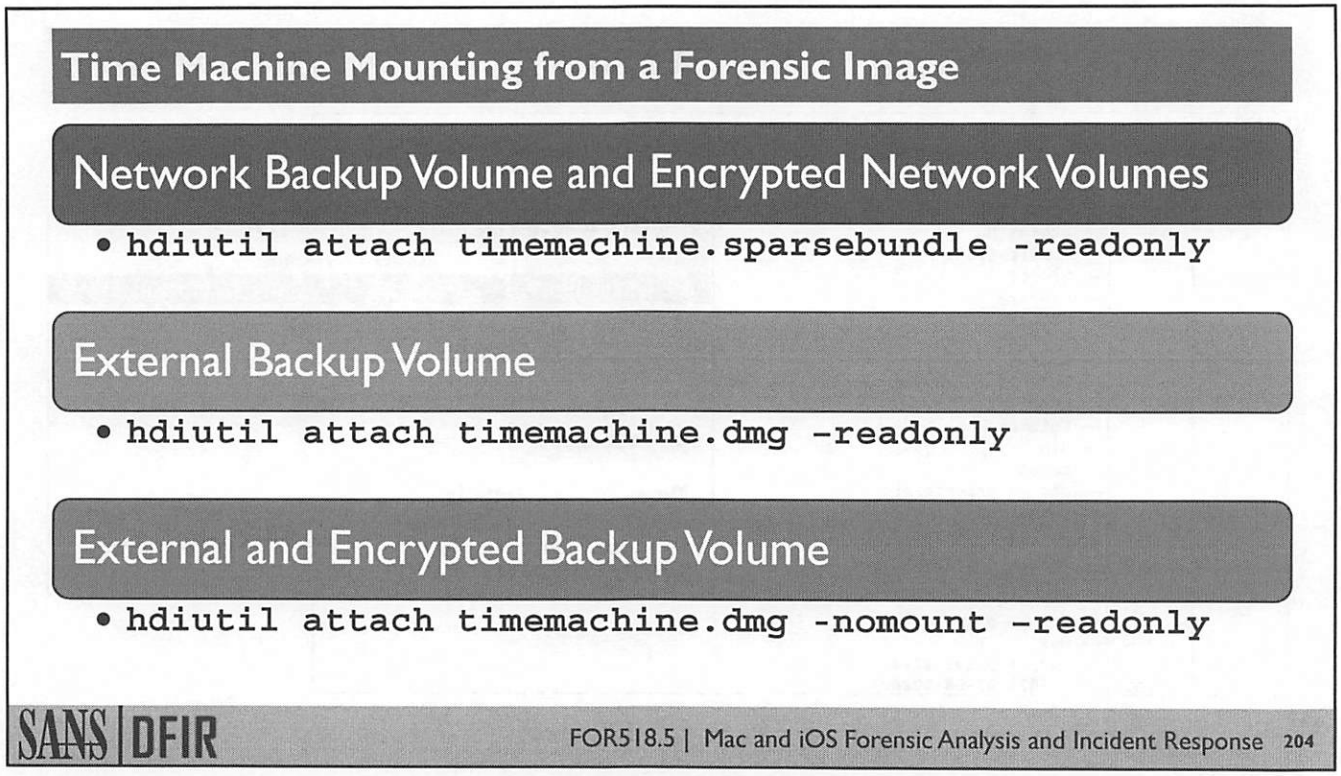

The `/.MobileBackups` directory in the root of the system volume contains a slightly different format than the `MobileBackups` volume. It will only contain those files that have been changed with no hard links to the rest of the file system. Instead of "`nibble.blah`" as in previous examples, it uses the more generic "`Computer`".



The mounted `MobileBackups` volume is a "`mtmfs`" volume, meaning it is a `Mobile Time Machine File System`.

The `tree` output in the screenshot above shows that the backup uses the same format as the normal Time Machine backups. It includes hard links to the full file system.

The `tree` command is not installed by default but can be downloaded from `http://mama.indstate.edu/users/ice/tree/` or installed via MacPorts, HomeBrew, or Fink.



We can use the `hdiutil attach` command to mount these Time Machine images, using the `-readonly` flag to be forensically sound.

This allows us to view the files inside the images as the system would have seen them. We can view all the files, hard links, and various snapshots. We can then use native OS X tools to start analyzing these files.

Each command uses nearly the same format, but the parameters have been adapted for the particular image type.

Time Machine: `tmutil uniquesize`

```
nibble:Dade's Mac sledwards$ tmutil uniquesize /Volumes/TIMEMACHINE/Backups.backupdb/Dade's\ Mac/*
171.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354
155.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-210636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-220636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-230638
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-000639
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-010641
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-020642
152.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-030645
152.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-040646
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-050649
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-060650
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-070652
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-080654
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-090656
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-100657
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-110700
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-120701
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-130703
152.7K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-140704
156.0K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-150707
157.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-160708
157.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-160708
```


The `tmutil` command-line utility interacts with Time Machine backups and may have to run with root privileges. We can use this utility to gather information about the backups and compare snapshots.

This utility can also be used on a system to change Time Machine preferences, start backups, restore files, or delete backups.

The verb `uniquesize` used with the `tmutil` utility will show the unique sizes of each snapshot, minus the hard-linked data. This is the true size of each snapshot backup.

Reference:
`tmutil` Man Page

```
nibble:Dade's Mac sledwards$ tmutil uniquesize /Volumes/TIMEMACHINE/Backups.backupdb/Dade's\ Mac/*
171.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354
155.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-210636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-220636
152.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-230638
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-000639
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-010641
152.2K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-020642
152.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-030645
152.3K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-040646
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-050649
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-060650
152.4K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-070652
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-080654
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-090656
152.5K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-100657
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-110700
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-120701
152.6K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-130703
152.7K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-140704
156.0K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-150707
157.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-160708
157.1K   /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-21-160708
```

Time Machine: `tmutil calculatedrift`

```
nibble:Backups.backupdb sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb
nibble:Backups.backupdb sledwards$ tmutil calculatedrift Dade's\ Mac/

2013-06-20-193354 - 2013-06-20-200636

-------------------------------------
Added:          1.3K
Removed:        0B
Changed:        166.1K


2013-06-20-200636 - 2013-06-20-210636

-------------------------------------
Added:          0B
Removed:        0B
Changed:        151.8K


2013-06-20-210636 - 2013-06-20-220636

-------------------------------------
Added:          0B
Removed:        0B
Changed:        275.2K
```

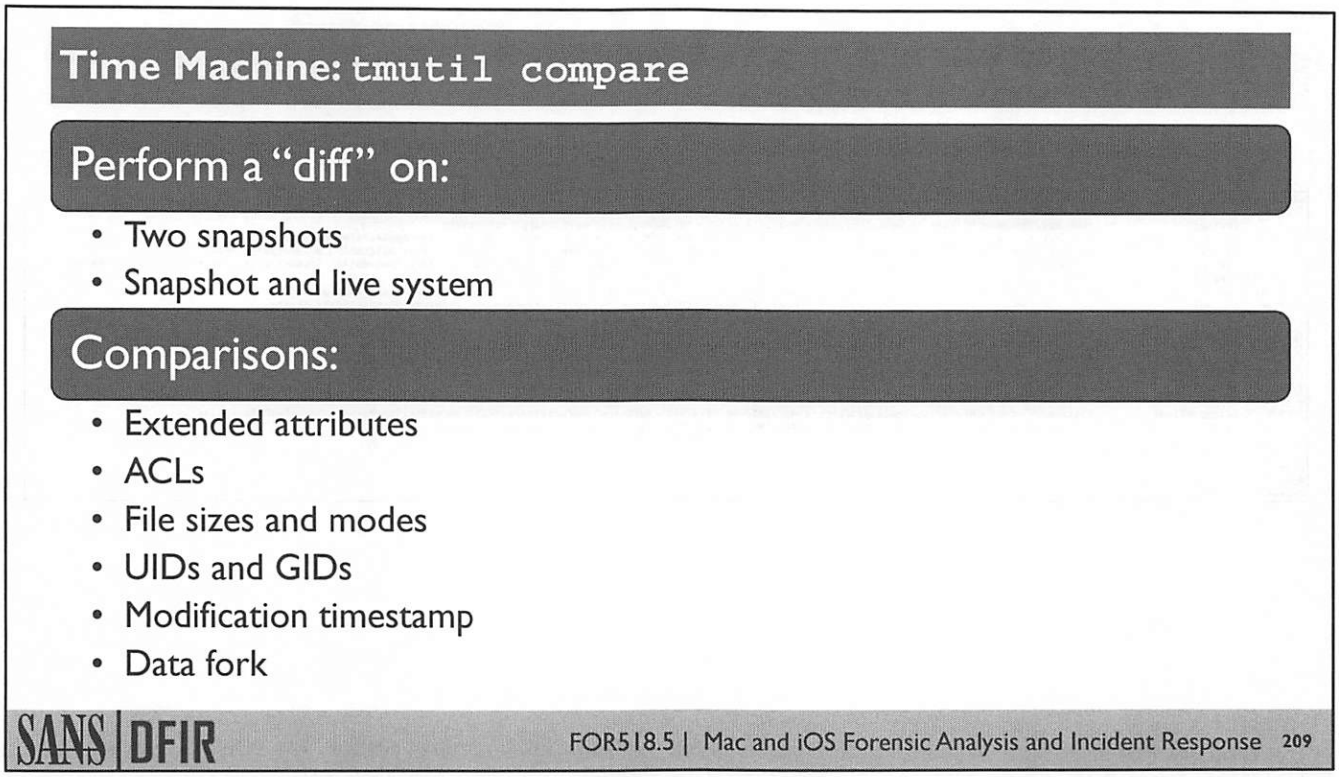

The verb `calculatedrift` used with the `tmutil` utility will show the size of changes between each snapshot. Each calculation includes the amount of data added, removed, and/or changed.

For example, between the snapshots `2013-06-20-193354` and `2013-06-20-200636`, 1.3K was added, nothing was removed, and 166.1K was changed. The "`tmutil calculatedrift`" command is used on the Machine Directory, in this example "`Dade's Mac`". This command will show the changes for each snapshot in this directory. At the completion of the command, it will show "`Drift Averages`", or averages of each addition, removal, or change.

This command is useful to determine which snapshot may have the most data additions, deletions, and changes on a system. Perhaps a large amount of data was removed from the system. This command is a quick test to determine which two snapshots to examine first to see what data was removed.

Reference:
`tmutil` Man Page

```
nibble:Backups.backupdb sledwards$ pwd
/Volumes/TIMEMACHINE/Backups.backupdb
nibble:Backups.backupdb sledwards$ tmutil calculatedrift Dade's\ Mac/

2013-06-20-193354 - 2013-06-20-200636

----------------------------------------
Added:          1.3K
Removed:        0B
Changed:        166.1K


2013-06-20-200636 - 2013-06-20-210636

----------------------------------------
Added:          0B
Removed:        0B
Changed:        151.8K


2013-06-20-210636 - 2013-06-20-220636

----------------------------------------
Added:          0B
Removed:        0B
Changed:        275.2K
```



The verb `compare` used with the `tmutil` utility will perform a diff between snapshots or a snapshot and the live system.

By default, it will compare the following values:

- File Size
- File Mode
- UID
- GID
- Modification Time

Reference:
`tmutil` Man Page

Time Machine: `tmutil compare` (Output)

```
nibble:Dade's Mac sledwards$ tmutil compare 2013-06-20-193354 2013-06-20-200636
! 1818    (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/.com.apple.backupd.mvlist.plist
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Library/Preferences
! 1.3K    (size, mtime)        /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Library/Preferences/com.apple.TimeMachine.plist
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Application Support/CrashReporter
! 3.3K    (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Application Support/CrashReporter/Intervals_00000000-0000
-1000-8000-000C29996BBE.plist
! 108.0K  (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db
! 32.0K   (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db-shm
! 4.1K    (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db-wal
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences
! 9.0K    (size, mtime)        /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences/com.apple.finder.plist
! 8.0K    (size, mtime)        /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences/com.apple.sidebarlists.plist
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Volumes
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db
+ 1.3K                         /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/.TimeMachine.Results.plist
! 292B    (size, mtime)        /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/com.apple.TimeMachine.SnapshotDates.plist
!         (mtime)              /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/run

---------------------------------------
Added:          1.3K
Removed:        0B
Changed:        166.1K
```


The default output for the `tmutil compare` command is shown in the screenshot above. Each change is labeled with one of the following marks in the first column:

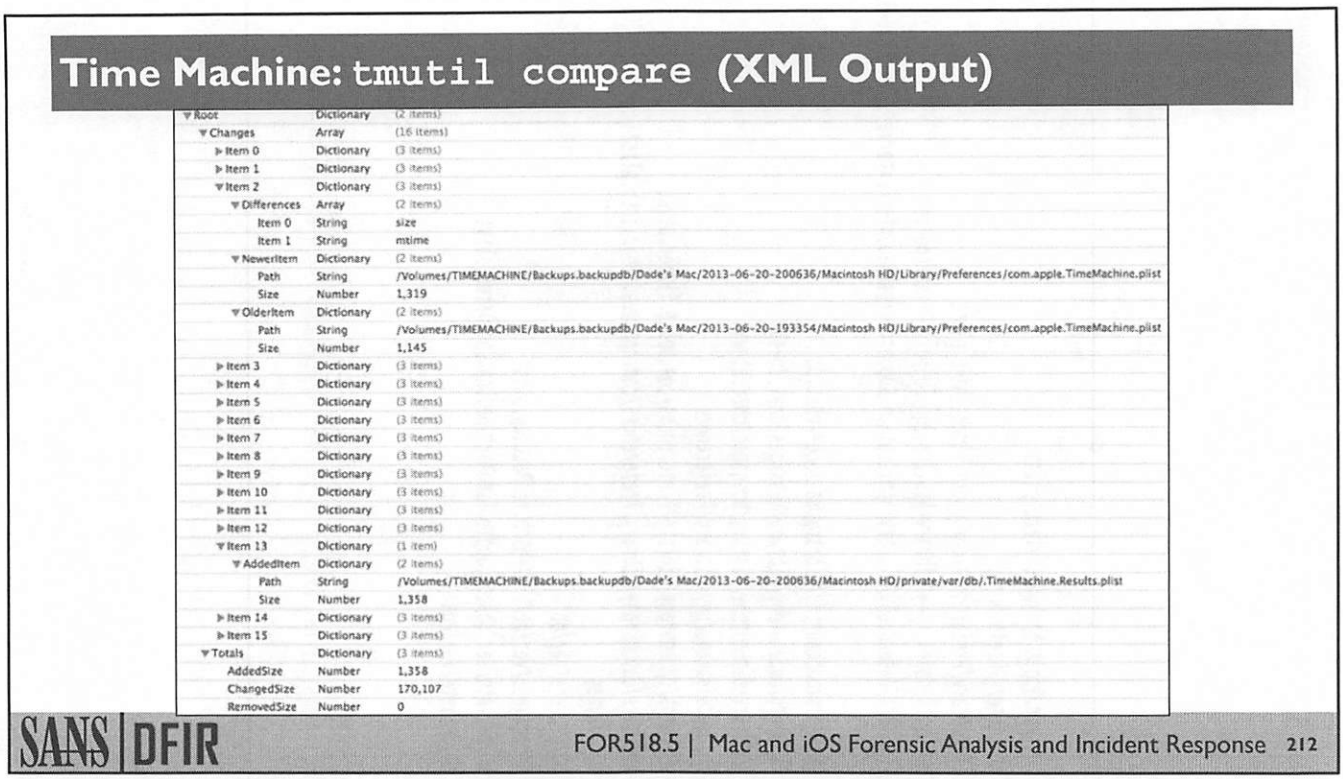
- ! Metadata Change
- + File Creation
- - File Removal

The second column shows the file size of the newest file.
The third column contains what metadata was changed.
The fourth column contains the file path to the specific file.

The last output contains the overall changes between the snapshots.

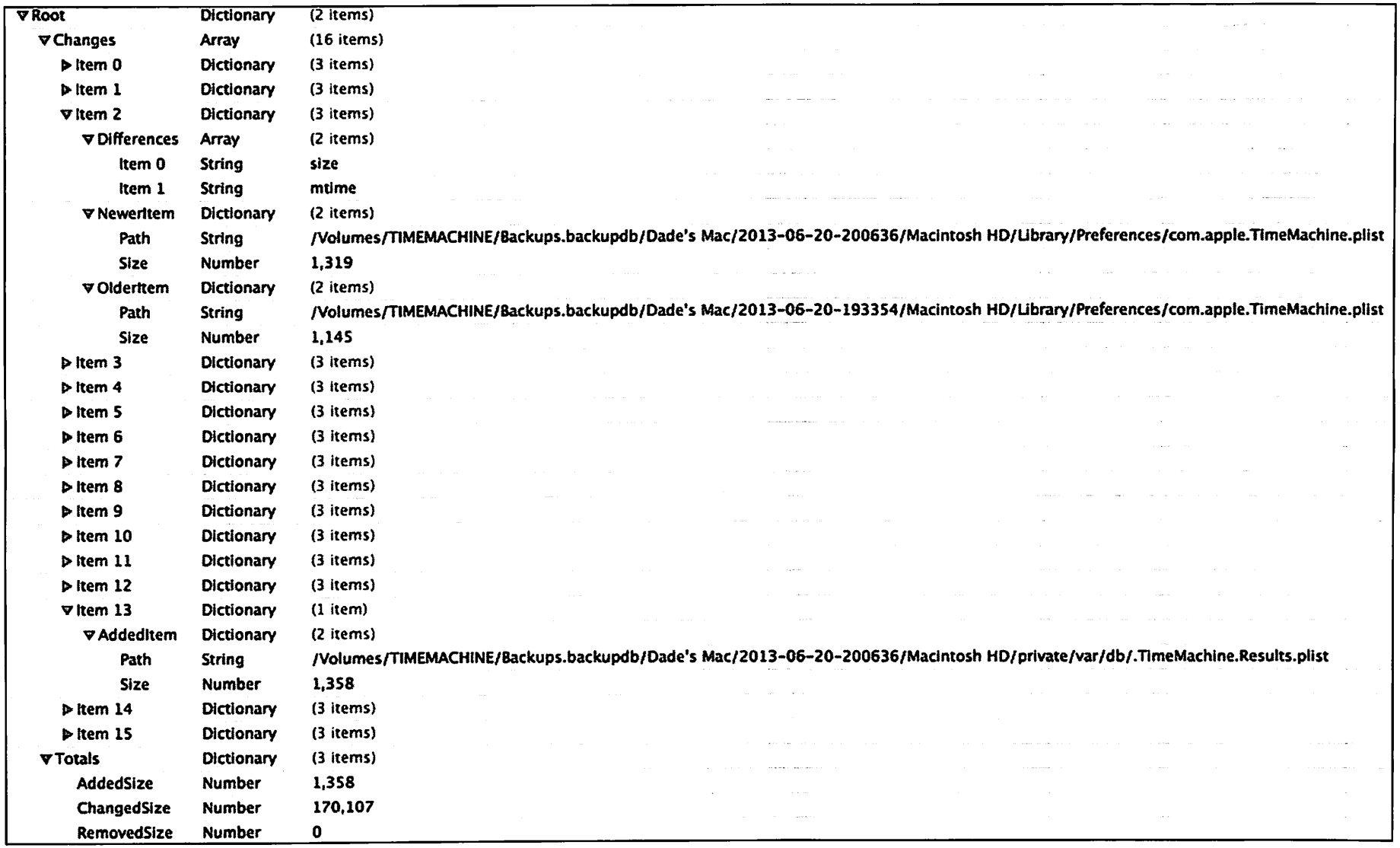
Reference:
`tmutil` Man Page

```
nibble:Dade's Mac sledwards$ tmutil compare 2013-06-20-193354 2013-06-20-200636
! 181B    (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/.com.apple.backupd.mvlist.plist
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Library/Preferences
! 1.3K    (size, mtime)       /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Library/Preferences/com.apple.TimeMachine.plist
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Application Support/CrashReporter
! 3.3K    (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Application Support/CrashReporter/Intervals_00000000-0000
-1000-8000-000C2999688E.plist
! 108.0K  (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db
! 32.0K   (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db-shm
! 4.1K    (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Messages/chat.db-wal
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences
! 9.0K    (size, mtime)       /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences/com.apple.finder.plist
! 8.0K    (size, mtime)       /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Users/dademurphy/Library/Preferences/com.apple.sidebarlists.plist
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Volumes
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db
+ 1.3K                       /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/.TimeMachine.Results.plist
! 292B    (size, mtime)       /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/com.apple.TimeMachine.SnapshotDates.plist
!         (mtime)             /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/run

--------------------------------

Added:          1.3K
Removed:        0B
Changed:        166.1K
```

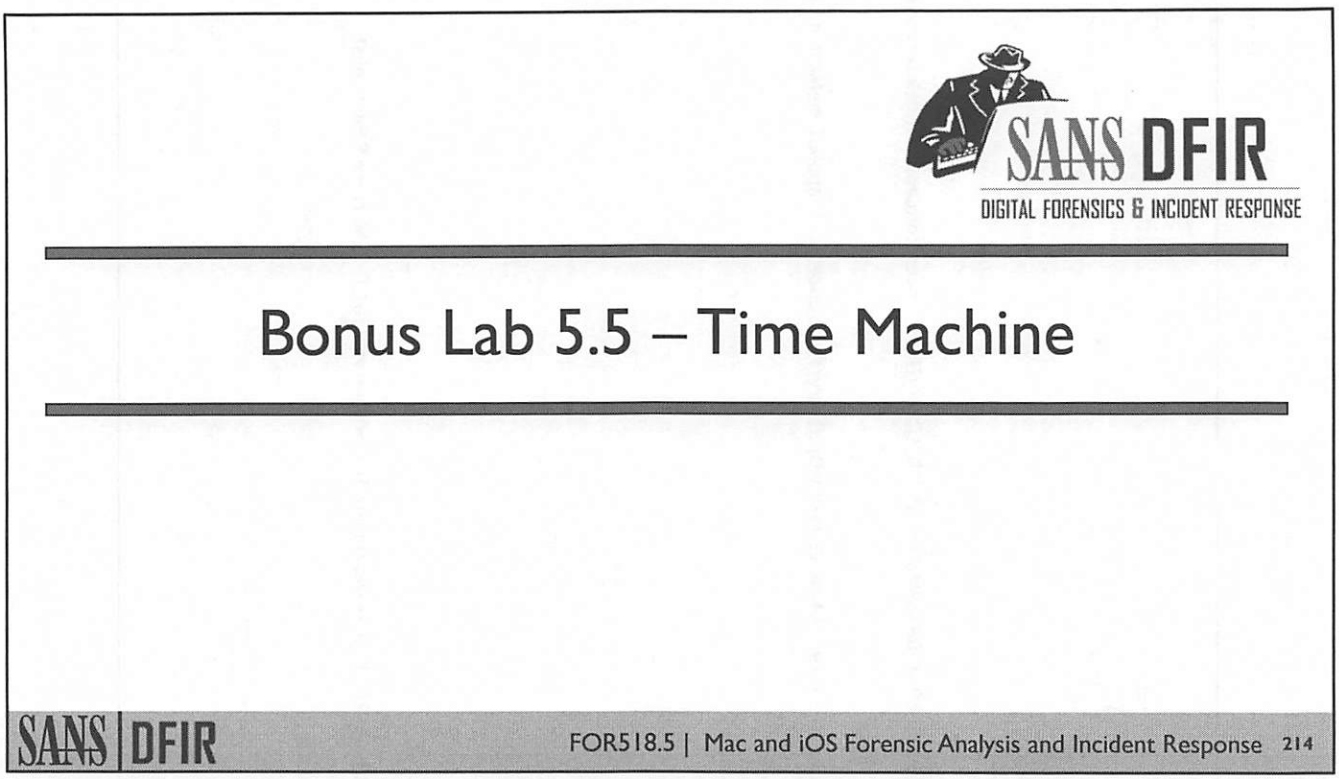


The `tmutil compare` command can also be output to an XML file for an easier and slightly more detailed view. This format can be viewed in Xcode, and file size changes are shown from the older file to the newer file.

Reference:
`tmutil` Man Page



▽Root	Dictionary	(2 items)
▽Changes	Array	(16 items)
▷Item 0	Dictionary	(3 items)
▷Item 1	Dictionary	(3 items)
▽Item 2	Dictionary	(3 items)
▽Differences	Array	(2 items)
Item 0	String	size
Item 1	String	mtime
▽NewerItem	Dictionary	(2 items)
Path	String	/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/Library/Preferences/com.apple.TimeMachine.plist
Size	Number	1,319
▽OlderItem	Dictionary	(2 items)
Path	String	/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-193354/Macintosh HD/Library/Preferences/com.apple.TimeMachine.plist
Size	Number	1,145
▷Item 3	Dictionary	(3 items)
▷Item 4	Dictionary	(3 items)
▷Item 5	Dictionary	(3 items)
▷Item 6	Dictionary	(3 items)
▷Item 7	Dictionary	(3 items)
▷Item 8	Dictionary	(3 items)
▷Item 9	Dictionary	(3 items)
▷Item 10	Dictionary	(3 items)
▷Item 11	Dictionary	(3 items)
▷Item 12	Dictionary	(3 items)
▽Item 13	Dictionary	(1 item)
▽AddedItem	Dictionary	(2 items)
Path	String	/Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/.TimeMachine.Results.plist
Size	Number	1,358
▷Item 14	Dictionary	(3 items)
▷Item 15	Dictionary	(3 items)
▽Totals	Dictionary	(3 items)
AddedSize	Number	1,358
ChangedSize	Number	170,107
RemovedSize	Number	0



This page intentionally left blank.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

"As usual, SANS courses pay for themselves by Day 2. By Day 3, you are itching to get back to the office to use what you've learned."
Ken Evans, Hewlett Packard Enterprise - Digital Investigation Services

SANS Programs
sans.org/programs

GIAC Certifications
Graduate Degree Programs
NetWars & CyberCity Ranges
Cyber Guardian
Security Awareness Training
CyberTalent Management
Group/Enterprise Purchase Arrangements
DoDD 8140
Community of Interest for NetSec
Cybersecurity Innovation Awards

Search SANSInstitute

SANS Free Resources
sans.org/security-resources

- E-Newsletters
  NewsBites: Bi-weekly digest of top news
  OUCH!: Monthly security awareness newsletter
  @RISK: Weekly summary of threats & mitigations
- Internet Storm Center
- CIS Critical Security Controls
- Blogs
- Security Posters
- Webcasts
- InfoSec Reading Room
- Top 25 Software Errors
- Security Policies
- Intrusion Detection FAQ
- Tip of the Day
- 20 Coolest Careers
- Security Glossary

SANS Institute
8120 Woodmont Avenue | Suite 310
Bethesda, MD 20814
301.654.SANS(7267)
info@sans.org