

518.5 Advanced Analysis Topics

SANS

518.5 Advanced Analysis Topics



Copyright © 2020, Sarah Edwards. All rights reserved to Sarah Edwards and/or SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

FOR518.5

Mac and iOS Forensic Analysis and Incident Response



FOR518 Section 5: Advanced Analysis Topics

© 2020 Sarah Edwards | All Rights Reserved | Version F02_01

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

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

Course Agenda

Section 1: Mac and iOS Essentials

Section 2: File Systems and System Triage

Section 3: User Data, System Configuration, and Log Analysis

Section 4: Application Data Analysis

Section 5: Advanced Analysis Topics

Section 6: Mac Forensic Challenge

This page intentionally left blank.



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Advanced Analysis Topics

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 3

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.

Section 5: Part I

Pattern of Life

This page intentionally left blank.

Pattern of Life Analysis

Applications	Battery Level and Charging Habits	Network Consumption and Usage	CarPlay Activity
Audio Output/Input	Location Habits	Device Lock Status and Methods	Various Configuration Items
Heart Rate	Steps and Distance	Camera/Flashlight Usage	User Activities

This page intentionally left blank.

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

```

1 SELECT
2 datetime(ZOBJECT.ZCREATIONDATE+978307200,'UNIXEPOCH') as 'ENTRY CREATION',
3 ZOBJECT.ZVALUESTRING AS 'BUNDLE ID',
4 CASE ZOBJECT.ZSTARTDAYOFWEEK
5   WHEN "1" THEN "Sunday"
6   WHEN "2" THEN "Monday"
7   WHEN "3" THEN "Tuesday"
8   WHEN "4" THEN "Wednesday"
9   WHEN "5" THEN "Thursday"
10  WHEN "6" THEN "Friday"
11  WHEN "7" THEN "Saturday"
12 END "DAY OF WEEK",
13 ZOBJECT.ZSECONDSFROMGMT/3600 AS "GMT OFFSET",
14 datetime(ZOBJECT.ZSTARTDATE+978307200,'UNIXEPOCH') as "START",
15 datetime(ZOBJECT.ZENDDATE+978307200,'UNIXEPOCH') as "END",
16 (ZOBJECT.ZENDDATE-ZOBJECT.ZSTARTDATE) as "USAGE IN SECONDS"
17 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 21:59:04	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
216	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

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-knowledgedb-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-knowledgedb>
- <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-outputter-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	3.016666666666667	181.0	count/min	_HKPrivateHeartRateContext	3637289
479577	2018-09-16 16:24:50	2018-09-16 16:24:50	5	3.033333333333333	182.0	count/min	HKMetadataKeyHeartRateMotionContext	3637294
479578	2018-09-16 16:24:50	2018-09-16 16:24:50	5	3.033333333333333	182.0	count/min	_HKPrivateHeartRateContext	3637294
479579	2018-09-16 16:24:53	2018-09-16 16:24:53	5	3.066666666666667	184.0	count/min	HKMetadataKeyHeartRateMotionContext	3637301
479580	2018-09-16 16:24:53	2018-09-16 16:24:53	5	3.066666666666667	184.0	count/min	_HKPrivateHeartRateContext	3637301
479581	2018-09-16 16:25:03	2018-09-16 16:25:03	5	3.116666666666667	187.0	count/min	HKMetadataKeyHeartRateMotionContext	3637310
479582	2018-09-16 16:25:03	2018-09-16 16:25:03	5	3.116666666666667	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	3.083333333333333	185.0	count/min	HKMetadataKeyHeartRateMotionContext	3637335
479586	2018-09-16 16:25:09	2018-09-16 16:25:09	5	3.083333333333333	185.0	count/min	_HKPrivateHeartRateContext	3637335

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-outputter-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 20:21:06	7	138.0	3637723
171	2018-09-16 20:21:06	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	200.604227876522	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	38.4893903959712	3637786
162	2018-09-16 21:09:58	2018-09-16 21:11:00	8	14.7879310355056	3637795

SANS | **DFIR**

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 10

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-outputter-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	176.379775510204	/media/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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 \$...)	214.529180997732	/media/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying
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/howPlaying

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 11

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-knowledgedb-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-knowledgedb>
<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-knowledgedb-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-knowledgedb>

<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>

Device Status – Aggregate Dictionary (iOS Physical) ADDataStore.db – Passcode Unlock (1 Week)

```

2  date(daysSince1970*86400,'unixepoch','utc') as day,
3  Key,
4  Value
5  from Scalars
6  where key like "%passcode%"
  
```

	day	key	value
1	2018-09-12	com.apple.springboard.lockscreen.passcodeUI.activationCount	9
2	2018-09-12	com.apple.passcode.PasscodeType	3
3	2018-09-12	com.apple.passcode.NumPasscodeEntered	1
4	2018-09-13	com.apple.springboard.lockscreen.passcodeUI.activationCount	7
5	2018-09-13	com.apple.passcode.PasscodeType	3
6	2018-09-13	com.apple.passcode.NumPasscodeEntered	1
7	2018-09-14	com.apple.springboard.lockscreen.passcodeUI.activationCount	15
8	2018-09-14	com.apple.passcode.PasscodeType	3
9	2018-09-14	com.apple.passcode.NumPasscodeEntered	3
10	2018-09-15	com.apple.passcode.PasscodeType	3
11	2018-09-15	com.apple.springboard.lockscreen.passcodeUI.activationCount	10
12	2018-09-16	com.apple.passcode.PasscodeType	3
13	2018-09-16	com.apple.springboard.lockscreen.passcodeUI.activationCount	12
14	2018-09-17	com.apple.passcode.PasscodeType	0
15	2018-09-17	com.apple.springboard.lockscreen.passcodeUI.activationCount	7
16	2018-09-17	com.apple.passcode.NumPasscodeEntered	2

```

1  select
2  date(daysSince1970*86400,'unixepoch','utc') as day,
3  Key,
4  Value
5  from Scalars
6  where key like "%fingerprint%"
  
```

0 rows returned in 26ms from: select
 date(daysSince1970*86400,'unixepoch','utc') as day,
 Key,
 Value
 from Scalars
 where key like "%fingerprint%"

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

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>

Device Status – Battery Level – CurrentPowerlog.PLSQL

```
1 SELECT
2 DATETIME(TIMESTAMP, 'unixepoch') AS TIMESTAMP,
3 LEVEL,
4 ID AS "PLBATTERYAGENT_EVENTBACKWARD_BATTERYUI TABLE ID"
5 FROM
6 PLBATTERYAGENT_EVENTSBACKWARD_BATTERYUI
```

	TIMESTAMP	Level	BATTERYAGENT_EVENTBACKWARD_BATTERYUI TABLE
763	2018-09-13 01:31:41	85.0	51178
764	2018-09-13 01:37:26	84.0	51179
765	2018-09-13 01:42:41	84.0	51180
766	2018-09-13 01:48:42	84.0	51181
767	2018-09-13 01:54:18	84.0	51182
768	2018-09-13 01:59:36	84.0	51183
769	2018-09-13 02:04:42	84.0	51184
780	2018-09-13 02:09:48	83.0	51185
761	2018-09-13 02:15:08	82.0	51186
782	2018-09-13 02:20:08	81.0	51187
763	2018-09-13 02:25:08	79.0	51188
764	2018-09-13 02:30:08	79.0	51189
765	2018-09-13 02:35:28	77.0	51190
766	2018-09-13 02:40:38	77.0	51191
767	2018-09-13 02:48:00	78.0	51192

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-outputter-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, -76.6329456498558] [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: 39.1152105276425, -76.632867058339] [ALTITUDE: 30.5] [COURSE: 152.069168091] [SPEED: 15.9477777778] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
318	Location	[TIMESTAMP: 2018-09-16 04:03:50] [COORDINATES: 39.1150827819936, -76.632722606686] [ALTITUDE: 30.7] [COURSE: 152.069168091] [SPEED: 15.9477777778] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
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: 39.1149566073617, -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, -76.6328033291797] [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, -76.6324485294773] [ALTITUDE: 30.7] [COURSE: 156.159225464] [SPEED: 15.6391111111] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...
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:F7:20:8D:77-Audio-AudioMain-43483...
329	Location	[TIMESTAMP: 2018-09-16 04:03:55] [COORDINATES: 39.114432869862, -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] [ASSERTION 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] [ASSERTION ID: 38320] [ASSERTION NAME: com.apple.audio.sid:0x379e4db, Music[...
332	Location	[TIMESTAMP: 2018-09-16 04:03:56] [COORDINATES: 39.1142981795731, -76.6322885162874] [ALTITUDE: 30.2] [COURSE: 157.228683472] [SPEED: 16.5136666667] [HORIZONTAL ACCURACY: 5.0] [VERTICAL ACCUR...

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-outputter-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>

Device Location Data

iOS Frequent locations

Location Databases

- “routined”: iOS
- Cellular: iOS
- Wi-Fi: iOS and macOS

Data Access (Physical on iOS)

Python Scripts

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.

iOS Frequent/Significant Locations (“routined”)

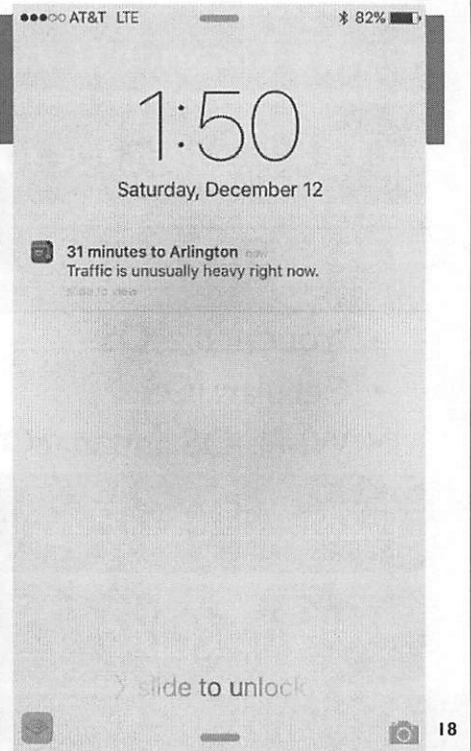
User Viewable (View Yours!):

- Settings ->
- Privacy ->
- Location Services ->
- System Services ->
- Frequent (or Significant with iOS 11+) Locations

Uses Location Services

More like “Frequent” and “Recent” Locations

“Frequent” algorithm is unknown



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

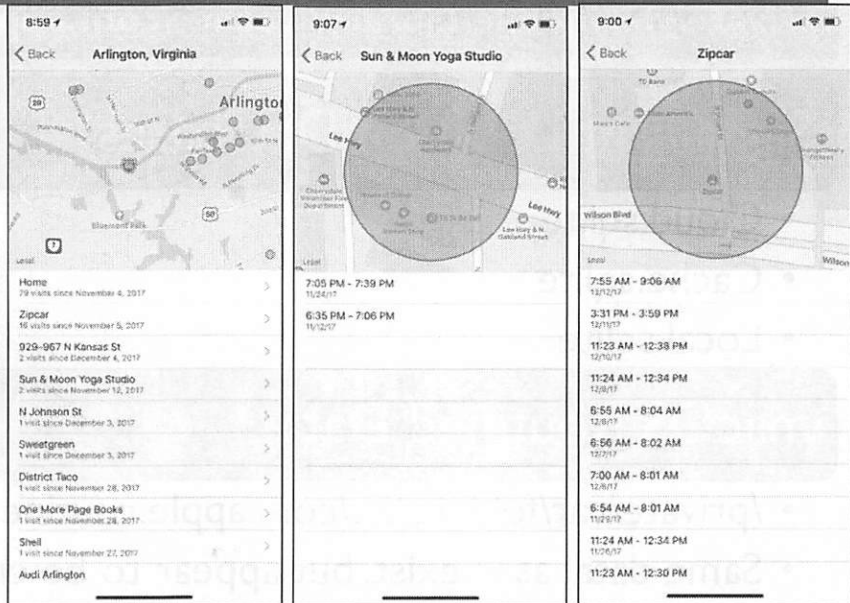
If enabled, this area will show a user's frequent and recent locations. While all locations are tracked, only those that get bubbled at the top of “frequent” or “recent” are shown in this list. This is not an inclusive list of everywhere the user has been.

iOS 11+ Significant Locations

Attempts to do more human-friendly reverse lookup

Not exactly accurate

- Zipcar every day?
 - No
- Yoga?
 - No



SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 19

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!

iOS 11+ and macOS 10.13+ Significant Locations
`/private/var/mobile/Library/Caches/com.apple.routined/`

Significant Location Databases

- Cloud.sqlite
- Cache.sqlite
- Local.sqlite

macOS Routined Data

- `/private/var/folders/.../com.apple.routined/Cache/`
- Same databases exist, but appear to be encrypted.

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

• ~ 1 Week – Granular Location (zrtclocationmo 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 12:51:40	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	38.8802153333333, -77.1258067333333	82.1	244.6	6.58488888888889	5.0	9.5	38.8802153333333	-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	38.8801638866667, -77.12594635	82.0	245.4	6.58488888888889	5.0	9.5	38.8801638866667	-77.12594635
2018-09-10 12:51:53	38.88014065, -77.1260116866667	82.0	246.0	6.17333333333333	10.0	19.0	38.88014065	-77.1260116866667
2018-09-10 12:51:54	38.88011795, -77.1260777666667	81.9	246.6	6.32786666666667	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_zrtclocationmo query from the APOLLO framework:

https://github.com/mac4n6/APOLLO/blob/master/modules/routined_cache_zrtclocationmo.txt

iOS 11+ Significant Locations – Cloud.sqlite (Visits) /private/var/mobile/Library/Caches/com.apple.routined/

VISIT ENTRY	VISIT EXIT	COORDINATES	PLACE ID	DATA POINT COUNT	LOCATION UNCERTAINTY	CONFIDENCE	VISIT CREATION	VISIT EXPIRATION
2018-05-10 16:18:40	2018-05-10 15:40:45	38.9461177802301, -77.45288895050885	174	89	67.9447058415279	1.0	2018-05-13 22:23:03	2018-11-02 18:47:03
2018-05-10 22:31:44	2018-05-10 23:00:16	32.7102102783546, -117.167878699929	176	95	94.4828484664364	1.0	2018-05-13 22:23:03	2018-11-02 18:47:03
2018-05-10 23:12:37	2018-05-11 00:41:12	32.7332440073273, -117.160932499837	177	289	25.6705794403562	1.0	2018-05-13 22:23:03	2018-11-02 18:47:03
2018-05-11 00:47:50	2018-05-12 01:41:21	32.7097870014053, -117.167857156334	178	3201	92.6712677347042	1.0	2018-05-13 22:23:03	2018-11-02 18:47:03
2018-05-12 01:51:15	2018-05-12 06:38:59	32.71257590006648, -117.157589200354	178	900	18.5834761578099	1.0	2018-05-13 22:23:03	2018-11-02 18:47:03
2018-05-12 06:44:43	2018-05-14 01:34:37	32.7096731454362, -117.167943410551	179	5015	74.0478614987874	1.0	2018-05-16 19:58:03	2018-11-02 18:47:03
2018-05-14 14:23:26	2018-05-15 12:67:46	32.70855464859568, -117.16784291645	179	2243	53.9615671291596	1.0	2018-05-16 19:58:03	2018-11-02 18:47:03
2018-05-16 23:17:13	2018-05-17 00:58:11	32.7105374103947, -117.170010318925	182	196	31.4320094807356	1.0	2018-05-17 18:17:38	2018-11-02 18:47:03
2018-05-17 00:58:28	2018-05-17 01:06:02	32.7099911645229, -117.167973858423	179	30	79.3556369952138	1.0	2018-05-17 18:17:38	2018-11-02 18:47:03
2018-05-17 01:23:31	2018-05-17 03:53:20	32.7404729990053, -117.211841530342	180	305	54.7658580371281	1.0	2018-05-17 18:17:38	2018-11-02 18:47:03
2018-05-17 04:04:45	2018-05-17 13:40:38	32.7097976590555, -117.167522705068	179	827	54.804291067164	1.0	2018-05-17 18:17:38	2018-11-02 18:47:03

DEVICE CLASS	DEVICE MODEL	DEVICE NAME	LEARNED PLACE CREATION	LEARNED PLACE EXPIRATION	MAP ITEM CREATION	PLACE NAME BLOB (HEX)	PLACE GEO BLOB (HEX)
iPhone	D221AP	AKEL	2018-05-13 22:26:22	2018-11-02 18:47:03	2018-05-13 22:26:22	080112E1021A120948202E631A7943401180...	0AA5151002210080A10018001817DC5741F...
iPhone	D221AP	AKEL	2018-05-13 22:26:27	2018-11-02 18:47:03	2018-05-13 22:26:26	080112F50108AE4D109EF2C7EEB8C92A5E...	DAD513089EF2C7EEB8C92A5E4011000221...
iPhone	D221AP	AKEL	2018-05-13 22:26:30	2018-11-02 18:47:03	2018-05-13 22:26:28	0801129D0208AE4D109ACCD9A18F948381...	0A9930089ACCD9A18F9483814210002218...
iPhone	D221AP	AKEL	2018-05-13 22:26:27	2018-11-02 18:47:03	2018-05-13 22:26:26	080112F50108AE4D109EF2C7EEB8C92A5E...	DAD513089EF2C7EEB8C92A5E4011000221...
iPhone	D221AP	AKEL	2018-05-13 22:26:34	2018-11-02 18:47:03	2018-05-13 22:26:32	080112880208AE4D10C29B93B0B0ED8FF7...	0AB92308D29B93B0B0ED8FF76A10002218...
iPhone	D221AP	AKEL	2018-05-16 19:58:16	2018-11-02 18:47:03	2018-05-16 19:58:05	080112880208AE4D1086DBFC8ABE91DECE...	0A9B240886DBFC8ABE91DECF501100022...
iPhone	D221AP	AKEL	2018-05-16 19:58:16	2018-11-02 18:47:03	2018-05-16 19:58:05	080112880208AE4D1086DBFC8ABE91DECE...	0A9B240886DBFC8ABE91DECF501100022...
iPhone	D221AP	AKEL	2018-05-18 03:07:49	2018-11-02 18:47:03	2018-05-18 03:07:48	080112880208AE4D10C8B998DEDEF4CBAF...	DAD92808CB8998DEDEF4CBAFA201100022...
iPhone	D221AP	AKEL	2018-05-16 19:58:16	2018-11-02 18:47:03	2018-05-16 19:58:05	080112880208AE4D1086DBFC8ABE91DECE...	0A9B240886DBFC8ABE91DECF501100022...
iPhone	D221AP	AKEL	2018-05-16 19:58:23	2018-11-02 18:47:03	2018-05-16 19:58:18	080112E20208AE4D10C2F7C8BBD307C8EE...	DAB73508C2F7C8BBD307C8EE2810002218...
iPhone	D221AP	AKEL	2018-05-16 19:58:16	2018-11-02 18:47:03	2018-05-16 19:58:05	080112880208AE4D1086DBFC8ABE91DECE...	0A9B240886DBFC8ABE91DECF501100022...

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

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

ENTRY	EXIT	COORDINATES	LOI LATITUDE	LOI LONGITUDE	ZCONFIDENCE	ZLOCATIONUNCERTAINTY	ZDATAPointCOUNT
2017-12-24 12:08:38	2017-12-24 12:39:57	51.4962035467992, -0.185025051758601	51.4962035467992	-0.185025051758601	1.0	108.638781388553	170
2017-12-24 12:41:01	2017-12-24 14:48:58	51.4949196173921, -0.188393153689219	51.4949196173921	-0.188393153689219	1.0	40.217172559999	360
2017-12-25 16:35:43	2017-12-25 17:26:24	51.5111285862601, -0.128776722014607	51.5111285862601	-0.128776722014607	1.0	116.411268434902	854
2017-12-25 17:43:21	2017-12-25 17:54:13	51.5012398780666, -0.125427016206231	51.5012398780666	-0.125427016206231	1.0	124.578104507777	86
2017-12-25 17:54:14	2017-12-25 18:05:58	51.5007710541901, -0.120411612787734	51.5007710541901	-0.120411612787734	1.0	97.2467464905957	133
2017-12-26 14:54:18	2017-12-26 15:15:13	51.5227431471224, -0.158162289756376	51.5227431471224	-0.158162289756376	1.0	123.042484956792	146
2017-12-26 16:07:41	2017-12-26 16:33:49	51.5048416849563, -0.0794814280031346	51.5048416849563	-0.0794814280031346	1.0	102.487458517847	579
2017-12-27 12:44:09	2017-12-27 16:00:24	51.9969130498878, -0.741205149893858	51.9969130498878	-0.741205149893858	1.0	134.962077147989	735
2017-12-28 14:09:52	2017-12-28 14:34:21	51.4862810638629, -0.13465602973605	51.4862810638629	-0.13465602973605	1.0	126.158597410675	113
2017-12-28 14:41:25	2017-12-28 14:53:11	51.4892783586073, -0.128079	51.4892783586073	-0.128079			
2017-12-29 17:57:53	2017-12-29 20:07:35	51.5191918168237, -0.1267219	51.5191918168237	-0.1267219			
2017-12-30 10:52:43	2017-12-30 14:00:16	51.5138812263551, -0.0987079	51.5138812263551	-0.0987079			

PLACE CREATION DATE	EXPIRATION	PLACE NAME BLOB (HEX)	PLACE GEO BLOB (HEX)	VISIT LATITUDE	VISIT LONGITUDE
2017-12-28 00:33:27	2018-10-09 03:43:33	0801128B0208AE4D10293D049F3072BF49401165...	0ABE0F10002210080A10016001817DFCD91E...	51.4952036467992	-0.185025051758601
2017-12-28 00:33:27	2018-10-09 03:43:33	0801128B0208AE4D10CCCF8680F986808E...	0A832E08DCCF8680F986808ED001100022...	51.4949166173921	-0.188393153689219
2017-12-28 00:33:27	2018-10-09 03:43:33	080112E0011A12095F506DA96CC148401100...	0AE31110002210080A10016001817DF5402E...	51.5111285862601	-0.128776722014607
2017-12-28 00:33:27	2018-10-09 03:43:33	080112D50108AE4D109F9EBCDA90E7E4F13...	0ACE26089F9EBDC90E7E4F134100022180...	51.5012398780666	-0.125427016206231
2017-12-28 00:33:27	2018-10-09 03:43:33	080112E70108AE4D1080C7FBBCDB0F9DC2...	0AEE110880C7FBBCDB0F9DC2AE011000221...	51.5007710541901	-0.120411612787734
2017-12-28 00:33:27	2018-10-09 03:43:33	08011287021A1209A78C583FE9C24940110...	0ACE1210002210080A10016001817DF6231D...	51.5227431471224	-0.158162289756376
2017-12-28 00:33:27	2018-10-09 03:43:33	0801129C011A1209B94E8FFA69ECC4940110...	0AE60F10002210080A10016001817DFDC168...	51.5048416849563	-0.0794814280031346
2017-12-28 00:33:27	2018-10-09 03:43:33	080112970208AE4D108B9E81D6A290B29E...	0AC52508BB9E81D6A290B29E8F011000221...	51.9969130498878	-0.741205149893858
2017-12-28 23:58:55	2018-10-09 03:43:33	080112990208AE4D10F2A298F5B2DC9284...	0AB42A08F2A298F5B2DC9284C401100022...	51.4862810638629	-0.13465602973605
2017-12-29 23:58:55	2018-10-09 03:43:33	08011284011A1209A8F3C3ACA0BE4940110...	0A981010002210080A10016001817DC8FFB...	51.4892783586073	-0.128079149934536
2017-12-29 23:59:24	2018-10-09 03:43:33	080112E40108AE4D108D6C82BAECE39CBE...	0AC46508BD9C82BAECE39CBE901100022...	51.5191918168237	-0.126721979969261
2017-12-29 23:59:24	2018-10-09 03:43:33	0801128C0208AE4D1094CC8EB9B0E298AB...	0AEC1D084CC8EB9B0E298AB6E10002218...	51.5087564130181	-0.0778518288993163
2017-12-31 18:08:36	2018-10-09 03:43:33	080112F90108AE4D10E4D3FB7BFEBD9EE9...	0AAE2D08E4D3FB7BFEBD9EE9C01100022...	51.5138812263551	-0.0987079913160755



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

Cellular/Wi-Fi Locations (locationd) cache_encrypted*.db and lockCache_encrypted*.db

Data Retention:
~1 Week
(Varies per Table)

Timestamps:
Accurate*

GPS Accuracy:
Within General Area

Many Other Tables:

- CDMA
- SCDMA
- LTE
- WIFI
- "Indoor"
- Application/"WTW"

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

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

SANS DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 24

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.

Wi-Fi Locations (iOS and macOS): iOS—cache_encryptedB.db macOS—cache_encryptedA.db / lockCache_encryptedA.db

```

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

```

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

Data Retention:
~4 Days

Timestamps:
Accurate*

GPS Accuracy:
Within General Area

MAC Address:
Stored in Base10

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 iPhone 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.

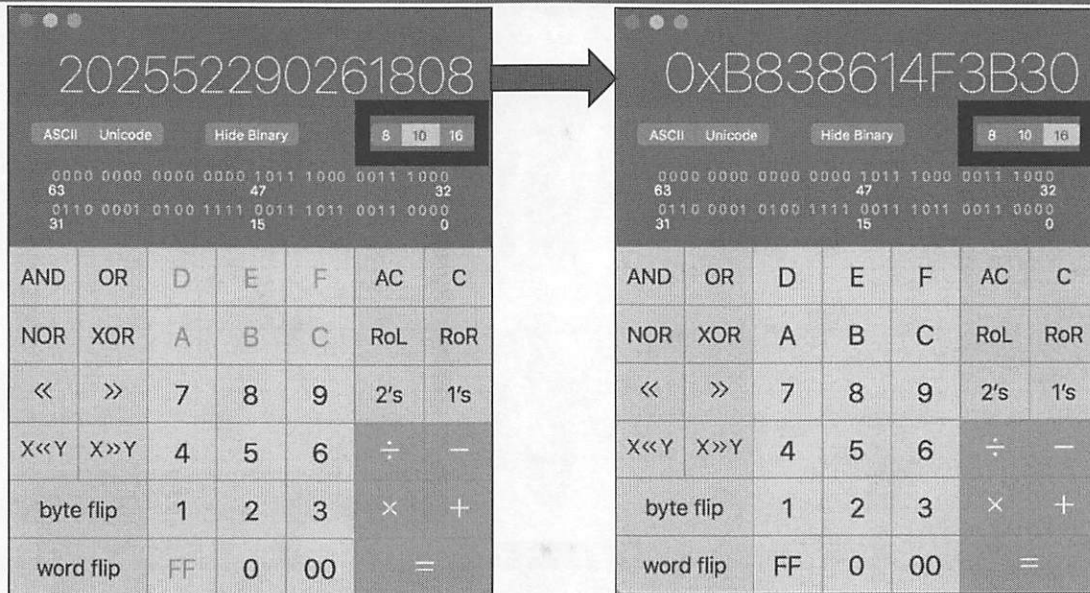
The rows in this table have been ordered by the “Timestamp” column.

```

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

```


Wi-Fi Locations: cache_encryptedB.db



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).

Network Search

General Search Network Detail

Query for networks

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

Search Radius Tolerance(±) degrees: 0.010

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

SSID / Network Name (exact match): foobar

SSID / Network Name (wildcards: % and _): foobar%

Last Observed: 200109251745

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

Addresses are for the U.S. only (2062 Census data)

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

Query Reset

¹ SSID cannot start with a wildcard. % means zero-or-more characters. _ means a single character.

<< showing records 1 to 1 >>

Map	Net ID	SSID	Name	Type	First Seen	Most Recently	Crypto	Est. Lat	Est. Long	Channel	Bcn Int.	QoS	Found by Me	Free Pay	Comment
map	B8:3B:61:4F:3B:30	OAS_OPEN		infra	2014-10-05 15:51:05	2015-03-09 14:45:51		38.89280819	-77.03944397	1		2			add comment

SANS DFIR FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 28

Using the hex format of the MAC address, we can now search for it on services like wagle.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:
[Wagle.net](https://wagle.net)

← → ↻ <https://wagle.net/search>

View Uploads Info Stats Tools

Network Search

General Search Network Detail

Query for networks

Latitude: 47.25264 to: 47.25265 Longitude: -87.256243 to: -87.259244

Search Radius Tolerance(+/- degrees): 0.010

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

SSID / Network Name (exact match): foobar

SSID / Network Name (wildcards¹: % and _): foobar%

Last Observed: 2001092517454

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

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

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

Query Reset

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

<< showing records 1 to 1 >>

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

Network Location

Click for interactive map

Lab 5.1

Pattern of Life

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.



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Section 5: Part 2

Document Versions

This page intentionally left blank.

Versions

Document Versions

TextEdit, Preview, iWork, and other supported third-party apps

Creates a new version of a document

- Every time it is opened
- Every hour it is open
- On demand

Saves hourly versions for a day, daily versions for a month, and weekly versions for previous months

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: /.DocumentRevisions-V100/ or on iOS Physical: /private/var/

```
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
```

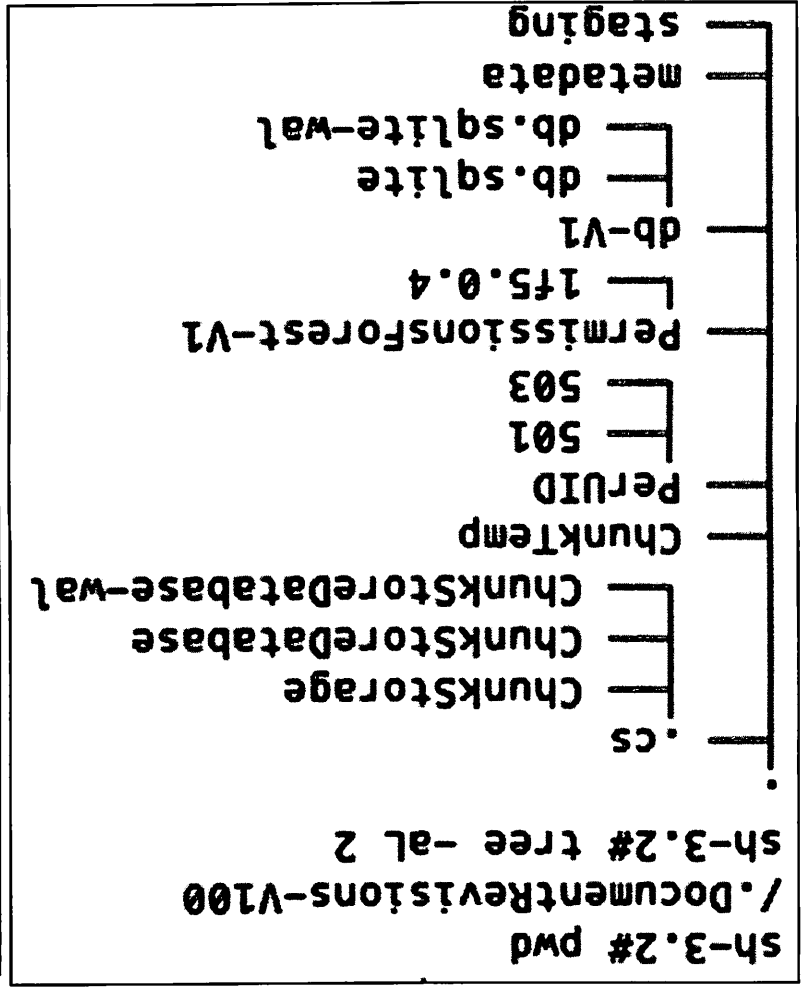
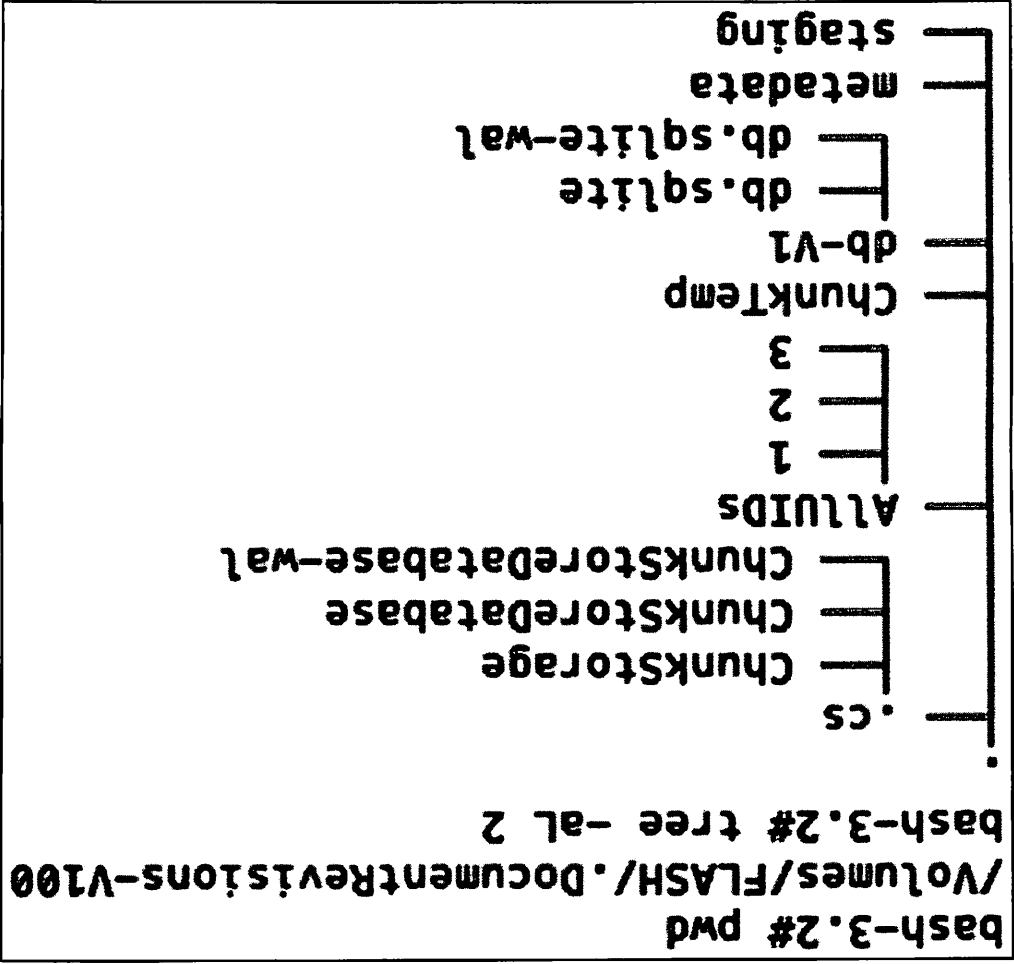
```
bash-3.2# pwd
/Volumes/FLASH/.DocumentRevisions-V100
bash-3.2# tree -aL 2
.
├── .cs
│   ├── ChunkStorage
│   ├── ChunkStoreDatabase
│   └── ChunkStoreDatabase-wal
├── AllUIDs
│   ├── 1
│   ├── 2
│   └── 3
├── ChunkTemp
├── db-V1
│   ├── db.sqlite
│   └── db.sqlite-wal
├── metadata
└── staging
```

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`.



Versions: UIDs /.DocumentRevisions—VI00/PerUID and AllUIDs

```
bash-3.2# tree AllUIDs/
AllUIDs/
├── 1
│   └── com.apple.documentVersions
│       ├── 1013CA57-AE06-4AB8-A7E2-BDEC038CE8DA.txt
│       ├── 58B3A232-4FEB-4806-BBA6-56656E53397D.rtf
│       ├── CD6201FF-698E-4E8F-891E-FAA5F0799B3D.rtf
│       └── FC61A18F-178C-4A9B-8DC9-D65E48ED67AD.rtf
├── 2
│   └── com.apple.documentVersions
│       ├── 30189267-5DE8-45FB
│       └── E983B555-E6A1-4240
├── 3
│   └── com.apple.documentVersions
│       ├── 6C567634-73F8-4803
│       ├── AA7CB143-5905-48AF
│       └── B7761245-7EE1-43E5
└── 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-BB60456250CD.rtf
    │   └── com.apple.ubiquity
    │       └── bookmark_format.C48AF86F-264F-4237-A76C-6C1F18714E7B.rtf
    └── 13
        └── com.apple.documentVersions
            └── DBD1E651-3D05-4CD3-833C-D54F17E7D012.plist
```

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 36

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-A7E2-8DEC038CE8DA.txt
│       ├── 58B3A232-4FEB-4B06-8BA6-56656E53397D.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 |NSPreservationRel|
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 |NSPreservationRel|
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

```

Versions: File Metadata com.apple.genstore.info Attribute

NSPreservationReason

- 1, 2, 10, 20, 30, 32, 40

NSDocumentPreviousSavedDate

- Previous Saved Date

com.apple.ubiquity.peername

- “iCloud” or system hostname

com.apple.ubiquity.moddate

- iCloud Modification Date

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`).

Versions: Versions Database /.DocumentRevisions-V100/db-V1/db.sqlite [1]

“files”
Table

Enter Field Values

1. file_row_id (INTEGER)	<input type="text" value="3"/>
2. file_name (TEXT)	<input type="text" value="testdoc.txt"/>
3. file_parent_id (INTEGER)	<input type="text" value="2"/>
4. file_path (TEXT)	<input type="text" value="/testdoc.txt"/>
5. file_inode (INTEGER)	<input type="text" value="239"/>
6. file_last_seen (INTEGER)	<input type="text" value="1379877723"/>
7. file_status (INTEGER)	<input type="text" value="1"/>
8. file_storage_id (INTEGER)	<input type="text" value="1"/>

db.sqlite

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

file_row_id	file_name	file_parent_id	file_path	file_inode	file_last_seen	file_status	file_storage_id
2	textfile.txt	2	/textfile.txt	229	1379877600	1	2
3	testdoc.txt	2	/testdoc.txt	239	1379877723	1	1
5	anothertestfile.rtf	270	./Trashes/501/anothertestfile.rtf	265	1379878122	1	3

41

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.

db.sqlite

Structure Browse & Search Execute SQL DB Setti

TABLE files Search Show All

file_row_id	file_name	file_parent_id	file_path	file_inode	file_last_seen	file_status	file_storage_id
2	textfile.txt	2	/textfile.txt	229	1379877600	1	2
3	testdoc.txt	2	/testdoc.txt	239	1379877723	1	1
5	anothertestfile.rtf	270	/.Trashes/501/anothertestfile.rtf	265	1379878122	1	3

Master Table (1)
Tables (4)
files
generations
sqlite_sequence
storage
Views (0)

Enter Field Values

1. file_row_id (INTEGER)
2. file_name (TEXT)
3. file_parent_id (INTEGER)
4. file_path (TEXT)
5. file_inode (INTEGER)
6. file_last_seen (INTEGER)
7. file_status (INTEGER)
8. file_storage_id (INTEGER)

Versions: Versions Database /.DocumentRevisions-V100/db-V1/db.sqlite [2]

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	E9838555-E6A1...	com.apple.documentVersions	AllUIDs/2/com...	3	1	1379877537	116
5	1	5883A232-4FEB...	com.apple.documentVersions	AllUIDs/1/com...	3	1	1379877586	349
6	1							
7	3							
8	3							
9	3							

“generations”
Table

Enter Field Values

1. generation_id (INTEGER)
2. generation_storage_id (INTEGER)
3. generation_name (TEXT)
4. generation_client_id (TEXT)
5. generation_path (TEXT)
6. generation_options (INTEGER)
7. generation_status (INTEGER)
8. generation_add_time (INTEGER)
9. generation_size (INTEGER)

43

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

Structure Browse & Search Execute SQL DB Settings

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

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	E9838555-E6A1...	com.apple.documentVersions	AllUIDs/2/com...	3	1	1379877537	116
5	1	5883A232-4FEB...	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	87761245-7EE1...	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

- generation_id (INTEGER)
- generation_storage_id (INTEGER)
- generation_name (TEXT)
- generation_client_id (TEXT)
- generation_path (TEXT)
- generation_options (INTEGER)
- generation_status (INTEGER)
- generation_add_time (INTEGER)
- generation_size (INTEGER)

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 5883A232-4FEB-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

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 5883A232-4FEB-4B06-BBA6-56656E53397D.rtf
-r--r--r--@ 1 _unknown _unknown 331 Sep 22 13:28 CD6201FF-698E-4E8F-891E-FAA5F079983D.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-FAA5F079983D.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.

Versions: Chunk Store Database /.cs/ChunkStoreDatabase

Structure Browse & Search Execute SQL DB Settings

ChunkStoreDatabase

Master Table (1)
Tables (8)
CSChunkTable

ct_rowid	cid	ct_lid	ft_rowid	offset	dataLen	refCount	timeStamp
1	X'01DC24D271C9A63761C96181431614019C55249...	1	0	356	1	1379870929	
2	X'013DAA8E8F0D5F1AD8F1EFD2FBDD8676DBC7AF503B'	1	356	365	1	1379870959	

TABLE CSChunkTable Search Show All Add

1. ct_rowid (INTEGER) 1 871117
2. cid (BLOB) X'01DC24D271C9A63761C96181431614019C552 877537
3. ct_lid (BIGINT) Null 877586
4. ft_rowid (BIGINT) 1 877723
5. offset (BIGINT) 0 878067
6. dataLen (INTEGER) 356
7. refCount (INTEGER) 1
8. timeStamp (BIGINT) 1379870929
9. location (INTEGER) 115
10. key (BLOB) Null

SANS DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 48

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

ChunkStoreDatabase

Structure Browse & Search Execute SQL DB Settings

Master Table (1)
Tables (8)
CSChunkTable

CSDatabaseVersion
CSRegisteredFileTable
CSStorageChunkListTable
CSStorageFileTable
CSStoragePendingChunksTable
CSStoragePendingFileChunkListTable
sqlite_sequence

TABLE CSChunkTable Search Show All Add

ct_rowid	cid	ct_lid	ft_rowid	offset	dataLen	refCount	timeStamp
1	X'01DC24D271C9A63761C96181431614019C55249...		1	0	356	1	1379870929
2	X'013DAA8E8F0D5F1AD8F1EFD2FBDD8676DBC7AF5038'		1	356	365	1	1379870959
3	X'0167D87D3712E049D652624F951E3CF9290A875648'		1	721	126	1	1379871117
4	X'014170310684A04C4C2B80EC2B8F0A7EFE24A2F973'		1	847	141	1	1379877537
5	X'018A08F34BA631B74F3B855C00A641114EFA43CDF1'		1	988	374	1	1379877586
6	X'016F2ADE9F58586C7839A96C32C202566042DA3B03'		1	1362	72	1	1379877723
7	X'0160BDBC0030F6288E1B2E1DC110B2B68D3404CABD'		1	1434	349	1	1379878067

1. ct_rowid (INTEGER)

2. cid (BLOB)

3. ct_lid (BIGINT)

4. ft_rowid (BIGINT)

5. offset (BIGINT)

6. dataLen (INTEGER)

7. refCount (INTEGER)

8. timeStamp (BIGINT)

9. location (INTEGER)

10. key (BLOB)

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	00 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\font\swiss\fch
0100	61 72 73 65	74 30 20 48	65 6C 76 65	74 69 63 61	38 70 00 7B	arset0 Helvetica;}.{
0120	5C 63 6F 6C	6F 72 74 62	6C 38 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 38 70	00 5C 60 61	een255\blue255;}.{\ma
0160	72 67 6C 31	34 34 30 5C	60 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	eww10000\viewh8400\w
0200	69 65 77 68	69 6E 64 30	00 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 00	00 5C 66 30	5C 66 73 32	rdinatural..{\font\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 60 65	6E 74 2E 5C	00 5C 00 70	00 00 01 60	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	38 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	00 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\font\swiss\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	i	cpg1252\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	o	nttbl\f0\fwiss\fc
0100	61 72 73 65	74 30 20 48	65 6C 76 65	74 69 63 61	3B 7D 0D 7B	a	rset0 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	e	een255\blue255;}. \ma
0160	72 67 6C 31	34 34 30 5C	6D 61 72 67	72 31 34 34	30 5C 76 69	r	gl1440\margr1440\vi
0180	65 77 77 31	30 38 30 30	5C 76 69 65	77 68 38 34	30 30 5C 76	e	ww10800\viewh8400\
0200	69 65 77 6B	69 6E 64 30	0D 5C 70 61	72 64 5C 74	78 37 32 30	i	ewkind0.\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	4	0\tx5760\tx6480\tx7
0280	32 30 30 5C	74 78 37 39	32 30 5C 74	78 38 36 34	30 5C 70 61	2	00\tx7920\tx8640\pa
0300	72 64 69 72	6E 61 74 75	72 61 6C 0D	0D 5C 66 30	5C 66 73 32	r	dirnatural..\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	1	252\cocoartf1187\co
0420	63 6F 61 73	75 62 72 74	66 33 39 30	0D 7B 5C 66	6F 6E 74 74	c	oasubrtf390.{\fontt
0440	62 6C 5C 66	30 5C 66 73	77 69 73 73	5C 66 63 68	61 72 73 65	b	l\f0\fwiss\fc

Versions: Versions Database /.DocumentRevisions-VI00/db-VI/

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 C8 E3 95 9E E6 7C 00 68 D9 49	o..B.lduw..ëü....I.Qu.. ¢Éä..ei.kÜI
9C ED 01 98	50 48 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		.f..PK.....w.SA·Ï\H.....
00 00 00 00 00 01 00 00 00 00 00 00 00 73 74 6F 72 65 46 69 6C 65 6E 61 6D 65 50 48 01 02 14 00 14		storeFilenamePK.....
00 00 00 00 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		w.SA-6.ÿ....Y.....
45 00 00 00 6D 65 74 61 64 61 74 61 50 48 01 02 14 00 14 00 00 00 08 00 77 94 35 41 F4 76 65 F4 B8 08 00			E...metadataPK.....w.SAöve0»
00 F8 0E 00 00 07 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			.ø.....f...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		w.SAEIä.«... ..
00 00 CE 00 00 00 6D 6F 64 65 6C 50 48 01 02 14 00 14 00 00 00 08 00 77 94 35 41 E3 A0 88 7F 48 06 00 00			..Ï ..modeLPK.....w.SAä ..H...
68 C1 00 00 13 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			hÁ.....storeFilename
54 6F 44 61 74 61 50 48 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 08 35 56 DC 42 53 5C 41	7B 5C 72 74 66 31 5C 61 6E 73 69 5C 61 6E 73 69 63		..ëfC+#0µ. 5VÜB5\{\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\font
77 69 73 73 5C 66 63 68 61 72 73 65 74 30 20 48 65 6C 76 65 74 69 63 61 38 7D 0A 7B 5C 63 6F 6C 6F 72 74			wiss\fcharset0 Helvetica;} {\color
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 68 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 blahblahl
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}... .4HÏ-µ_3ñtaBe.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

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..β,íauw,éü...I.Qμ.. ³Ēā..æl.kÙI
9C ED 01 98	50 4B 01 02 14 00 14 00 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·Î\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.5Aôveó»..
00 F8 0E 00 00 07 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			.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 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 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 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\font
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			innatural \font\fs24 \cf0 blahblahl
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	ah}... .4HÎ-ª_μ3ÑiaBe.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

Lab 5.2

Document Versions

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.



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Section 5: Part 3

iCloud

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 56

This page intentionally left blank.

iCloud

Access data from anywhere on a Mac, iDevice, Windows, or web browser (10.7.2+ and iOS 5+)

- Documents, contacts, calendar, music, photos, email, more!

“Ubiquity” = Found everywhere

Data is backed up to Apple servers

- 5GB free, purchase up to 2TB

Each iCloud account has a numeric Person ID

Credentials

- Apple ID and password, two-factor, token



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.

iCloud: OS X and iOS



This page intentionally left blank.



This page intentionally left blank.

iCloud: Windows



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.


```
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]
```

iCloud Data Storage

iCloud Device Backups

- Device Settings
- Photos/Videos (if iCloud Photo Library not enabled)
- App Data and Configurations
- Messages
- Health Data
- HomeKit
- App Purchase History

iCloud Data

- iCloud Drive Data
- Documents
- Photos/Videos
- Contacts
- Calendars
- Bookmarks
- iCloud Mail
- Notes
- Etc.

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>

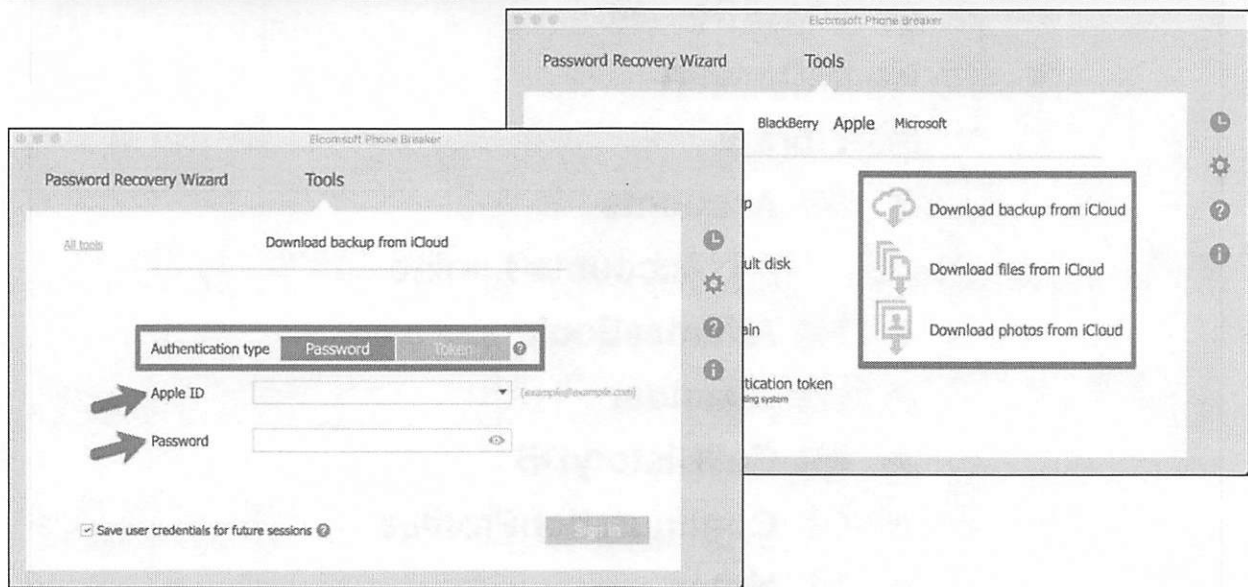
<http://www.oxygen-forensic.com/en/products/oxygen-forensic-detective/detective/cloud-data-extraction>

<http://www.cellebrite.com/Mobile-Forensics/Products/ufed-cloud-analyzer>

https://www.msab.com/products/xry/#_cloud



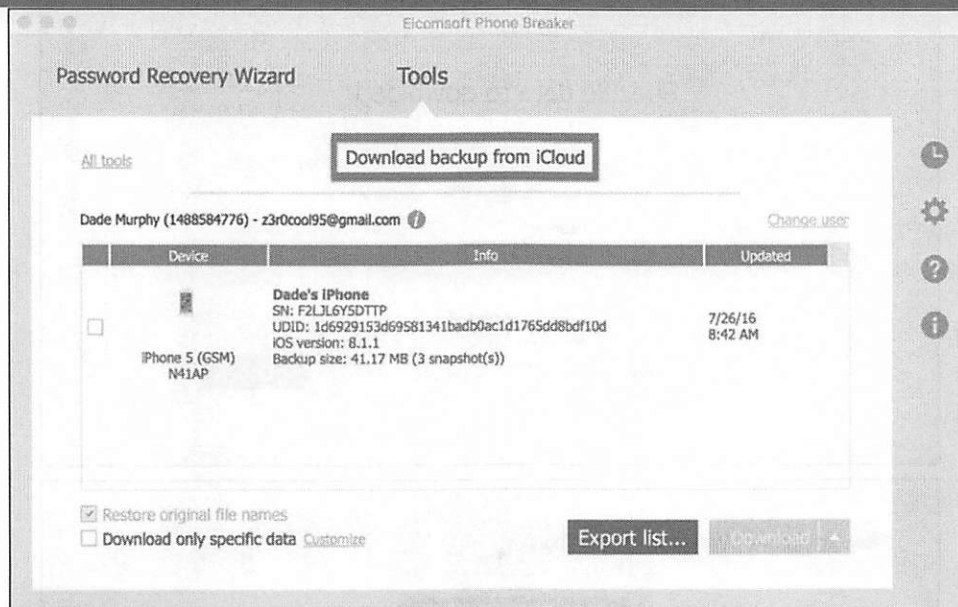
iCloud Data and Elcomsoft Phone Breaker [1]



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.

iCloud Data and Elcomsoft Phone Breaker [2]

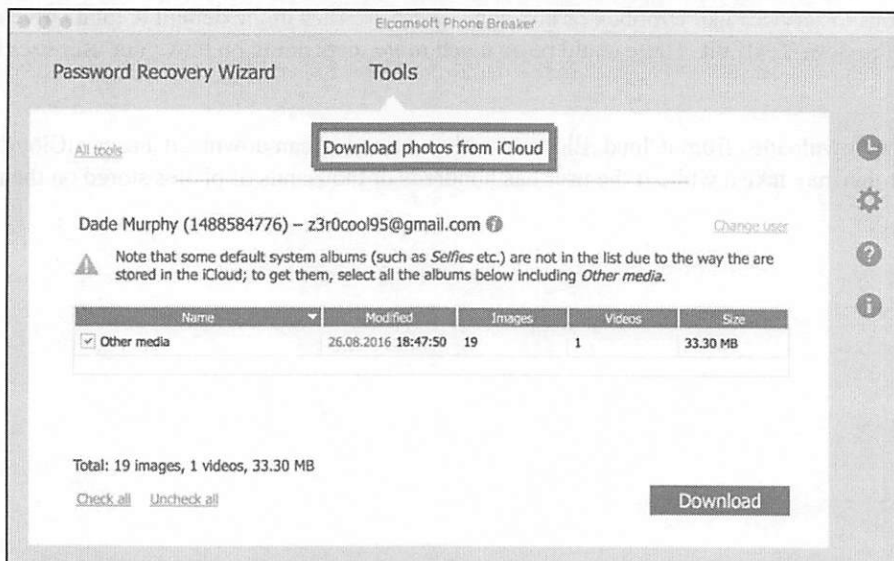
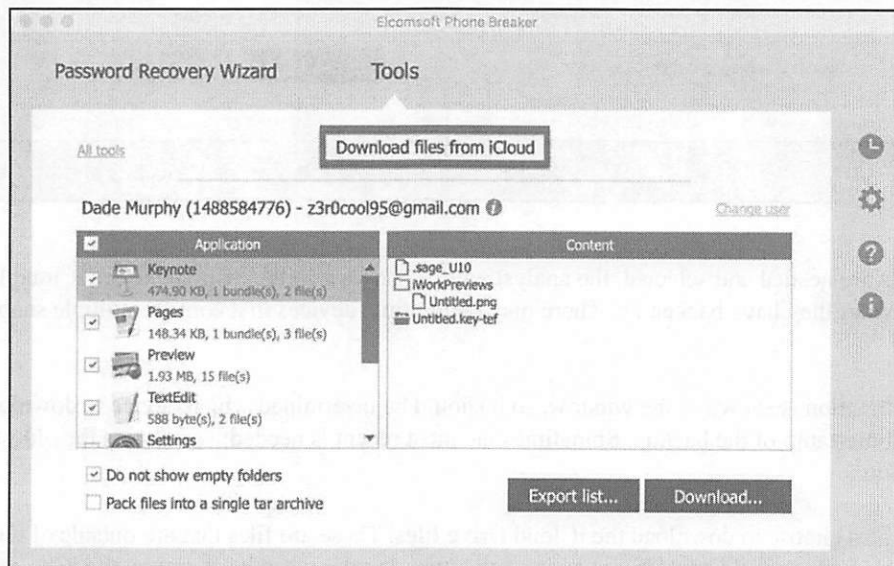
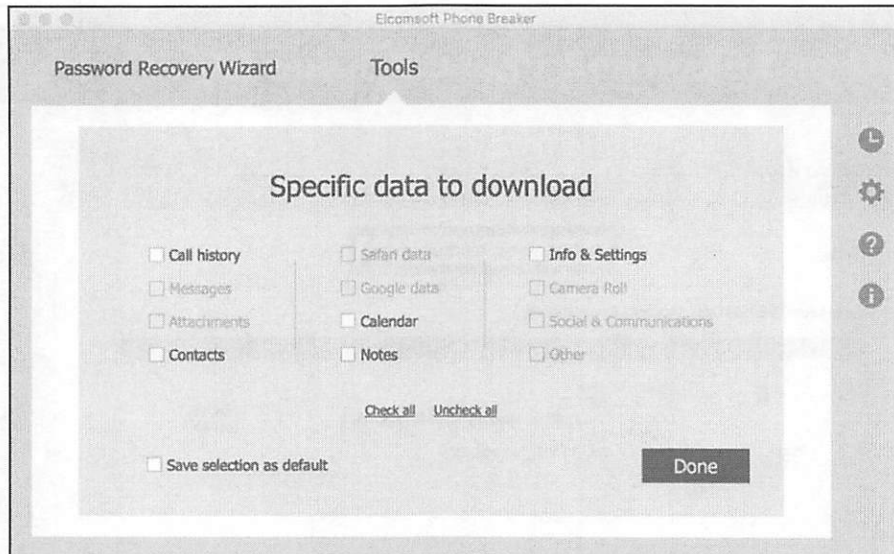


If iCloud Backups are needed and selected, the analyst will be shown a screen with the user's iCloud Backup files for whichever devices they have backed up. There may be multiple devices that contain multiple snapshots or backups of data.

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
```

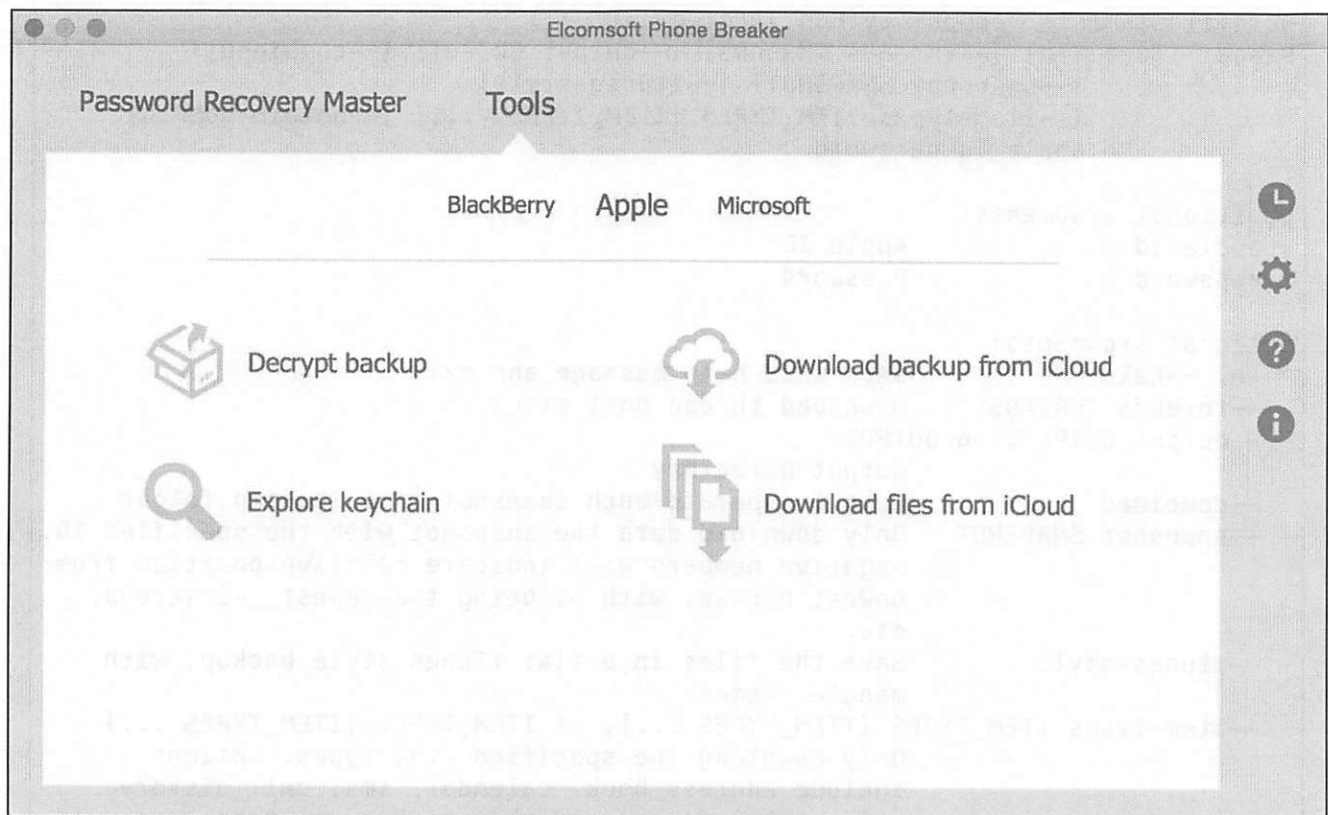
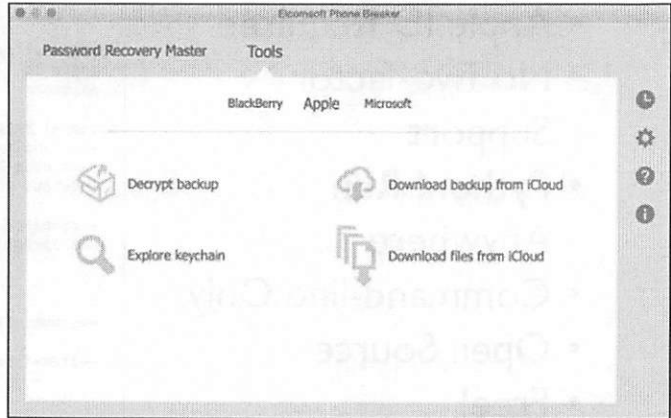
```
word:iloot-master oompa$ python iloot.py -h
usage: iloot [-h] [--threads THREADS] [--output OUTPUT] [--combined]
            [--snapshot SNAPSHOT] [--itunes-style]
            [--item-types ITEM_TYPES [ITEM_TYPES ...]] [--domain DOMAIN]
            apple_id password

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: Synced Preferences ~/Library/SyncedPreferences or ~/Library/Containers/...

Contains synced preferences for:

- Email
- Safari
- Wi-Fi
- Maps
- Stocks
- Weather
- Messages
- More and More with Every OS!

iOS Physical

- /private/var/mobile/Containers/
- /private/var/mobile/Library/SyncedPreferences/

```
Sarahs-MBP-5:~ oompa$ find ~ -ipath *SyncedPreferences
/Users/oompa/Library/SyncedPreferences
/Users/oompa/Library/SyncedPreferences/CommCenter.plist
/Users/oompa/Library/SyncedPreferences/suggested.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.cmfSyncAgent.plist
/Users/oompa/Library/SyncedPreferences/com.apple.sbd.plist
/Users/oompa/Library/SyncedPreferences/.dat.nosync2acd.DA462m
/Users/oompa/Library/SyncedPreferences/.dat.nosync5275.CvNwKk
/Users/oompa/Library/SyncedPreferences/.dat.nosync733a.BgXNlK
/Users/oompa/Library/SyncedPreferences/.dat.nosync67c7.08ybV9
/Users/oompa/Library/SyncedPreferences/com.apple.syncedpreferences.plist
/Users/oompa/Library/SyncedPreferences/com.apple.wifi.WiFiAgent.plist
/Users/oompa/Library/SyncedPreferences/.dat.nosync8351.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/recentst-com.apple.mail.recents.plist
/Users/oompa/Library/Containers/com.apple.ncplugin.weather/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.ncplugin.weather.plist
/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.Maps-com.apple.MapsSupport.history.plist
/Users/oompa/Library/Containers/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.plist
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/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.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.corerecents.recentst-com.apple.passbook.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.calendar.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.corerecents.map-locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.eventkit.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.facetime.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.mail.recents.plist
```

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/suggested.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.DA462m
/Users/oompa/Library/SyncedPreferences/.dat.nosync5275.CvNwKk
/Users/oompa/Library/SyncedPreferences/.dat.nosync733a.BgXNlK
/Users/oompa/Library/SyncedPreferences/.dat.nosync67c7.08ybV9
/Users/oompa/Library/SyncedPreferences/com.apple.syncedpreferences.plist
/Users/oompa/Library/SyncedPreferences/com.apple.wifi.WiFiAgent.plist
/Users/oompa/Library/SyncedPreferences/.dat.nosync8351.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/recentst-com.apple.mail.recents.plist
/Users/oompa/Library/Containers/com.apple.ncplugin.weather/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.ncplugin.weather.plist
/Users/oompa/Library/Containers/com.apple.Maps/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.Maps-com.apple.MapsSupport.history.plist
/Users/oompa/Library/Containers/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.plist
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/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.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences
/Users/oompa/Library/Containers/com.apple.corerecents.recentst-com.apple.passbook.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.calendar.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.corerecents.map-locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.eventkit.locations.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-com.apple.facetime.recents.plist
/Users/oompa/Library/Containers/com.apple.corerecents.recentst/Data/Library/SyncedPreferences/recentst-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.corerecent.recentSD/Data/Library/SyncedPreferences/recentSD-com.apple.mail.recents.plist

iOS:

/private/var/mobile/Library/SyncedPreferences/com.apple.cloudrecentSD.RecentSDAgent-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	oempa@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 bce0021b 000000
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	oempa@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	ree@sans.org
k	String	email
remotevalue	Data	<01196a02 3f79b21a 000000
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.

Synced Preferences: Wi-Fi—Synced Access Points

Key	Type	Value
▼ Root	Dictionary	(4 items)
versionid	String	FT--@RU=d
initialsync	Number	5
changecount	Number	30
▼ values	Dictionary	(37 items)
▶ FOR518	Dictionary	(3 items)
▶ Reagan National WiFi	Dictionary	(3 items)
▶ Parsons_Visitor	Dictionary	(3 items)
▶ ASUS	Dictionary	(3 items)
▶ SJ	Dictionary	(3 items)
▶ RitzCarlton_Guest	Dictionary	(3 items)
▶ Marriott_GUEST	Dictionary	(3 items)
▶ Hyatt Lobby	Dictionary	(3 items)
▶ scandic_easy	Dictionary	(3 items)

Key	Type	Value
▼ Reagan National WiFi	Dictionary	(3 items)
▼ value	Dictionary	(11 items)
WEP	Boolean	NO
enabled	Boolean	YES
UserDirected	Boolean	NO
added_by	String	miPhone5s
SSID_STR	String	Reagan National WiFi
IS_NETWORK_CUSTOMIZED	Boolean	NO
BSSID	String	0:1c:f6:60:45:30
added_at	String	Feb 9 2014 20:25:23
IS_NETWORK_CONFIGURED	Boolean	NO
IS_NETWORK_EAP	Boolean	NO
AP_MODE	Number	2
remotevalue	Data	<01b94900 576fd91a 00
timestamp	Number	450,457,431

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.

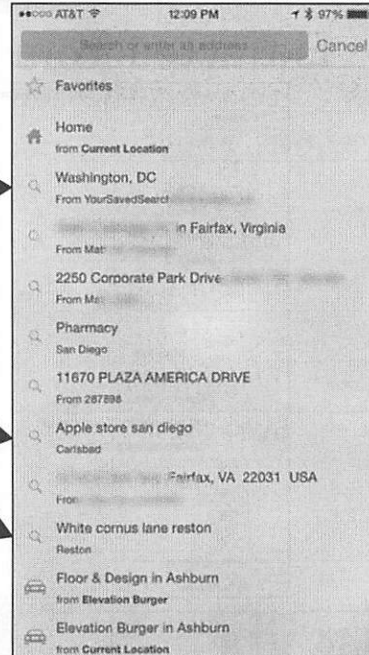
added_by: The device name that added this access point.

added_at: The time it was added. Warning: Not necessarily added when it was connected to.

Synced Preferences: Maps—Recent Addresses and Locations

Recent Addresses
(Extracted from Mail
emails: “From ...”)

Recent Locations and
Searches



75

- Recent Addresses (Extracted from Mail emails: “From ...”)
 - OS X:
~/Library/Containers/com.apple.corerecents.recentsd/Data/Library/SyncedPreferences/recentds-com.apple.corerecents.map-locations.plist
 - iOS:
/private/var/mobile/Library/SyncedPreferences/com.apple.cloudrecents.CloudRecentAgent-com.apple.corerecents.map-locations.plist
- Recent Locations and Searches
 - OS X:
 - ~/Library/SyncedPreferences/com.apple.Maps-com.apple.MapsSupport.history.plist
 - /Users/oempa/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 6726ed1a 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)

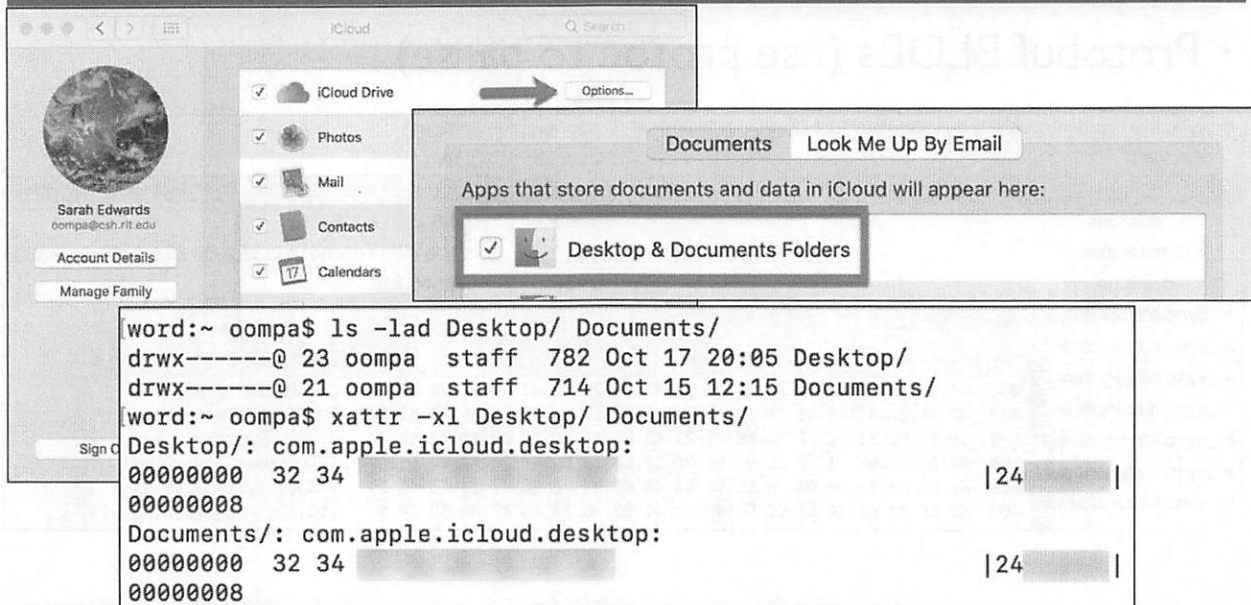
The screenshot shows a dictionary view for a synced preference item. The dictionary contains several fields: 'data' (Data), 'position' (Number), 'remotevalue' (Data), and 'timestamp' (Number). The 'data' field is expanded to show a Protobuf BLOB. Below the dictionary is a hex editor window titled 'Untitled' showing the raw bytes of the BLOB. The hex editor displays a grid of bytes with their corresponding ASCII values. A black arrow points from the 'data' field in the dictionary to the hex editor.

Field	Type	Value
value	Dictionary (2 items)	
data	Data	<08011224 44413232 30304138 2d414333 382d3442
position	Number	452,996,812.218939
remotevalue	Data	<01384800 cc2e001b 00000000 62706c69 73743030
timestamp	Number	452,996,812

Offset	Hex	ASCII
000	08 01 12 24 44 41 32 32 30 30 41 38 2D 41 43 33 38 2D 34 42	...\$DA2200A8-AC38-4B
020	42 33 2D 42 46 43 39 2D 38 44 45 39 31 46 41 35 45 39 31 34	B3-BFC9-8DE91FA5E914
040	19 63 0C 38 CC 2E 00 BB 41 21 63 0C 38 CC 2E 00 BB 41 32 3B	.c.8...A;c.8...A2;
060	0A 08 50 68 61 72 6D 61 63 79 12 09 53 61 6E 20 44 69 65 67	..Pharmacy.. San Dieg
080	6F 22 24 29 00 00 90 EB DE 58 40 40 31 00 00 4C 38 C8 4A 5D	o"\$).....X@01..L8.J]
100	C0 39 00 00 AA B2 CC 5C 40 40 41 00 00 8A F5 37 49 5D C0	.9.....\@A.....7I].

This page intentionally left blank.

iCloud Desktop and Documents Directories [10.12]



The screenshot shows the iCloud preferences window for Sarah Edwards (oompa@csh.fitt.edu). The 'iCloud Drive' section is highlighted, and the 'Options...' button is selected. A secondary window titled 'Documents Look Me Up By Email' is overlaid, showing a list of apps that store documents and data in iCloud. The 'Desktop & Documents Folders' app is checked and highlighted. Below this, a terminal window shows the following commands and output:

```
word:~ oompa$ ls -lad Desktop/ Documents/  
drwx-----@ 23 oompa  staff  782 Oct 17 20:05 Desktop/  
drwx-----@ 21 oompa  staff  714 Oct 15 12:15 Documents/  
word:~ oompa$ xattr -xl Desktop/ Documents/  
Desktop/: com.apple.icloud.desktop:  
00000000 32 34 [REDACTED] |24 [REDACTED] |  
00000008  
Documents/: com.apple.icloud.desktop:  
00000000 32 34 [REDACTED] |24 [REDACTED] |  
00000008
```

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: Mobile Documents ~/Library/Mobile Documents/

Local Storage of iCloud Data and Documents

- iWork, TextEdit, Notes, Preview, etc.

Extended Attributes of Mobile Documents Directory

- com.apple.ubd.prsid = iCloud Person ID

Native and Third-Party Apps

```

5719237FN3-net-whatsapp-WhatsApp
82J93X7T2S-com-apple-mobileiphoto
8YE23NZ357-com-kayak-travel
A4QBZ46HAP-com-gameoft-UNOfree
F3LWYJ7GM7-com-apple-mobilegarageband
F6266T9T75-com-apple-iMovie
UBHRZCX4PE-com-foodnetwork-tveverywhere
X6UDPZ1LVR-QSCSS29KB3-com-velyco-iDashb
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-shoobox
com-apple-system-spotlight
iCloud-com-apple-iBooks
iCloud-com-apple-iBooks-iTunesU
iCloud-com-apple-itunesu
iCloud-com-apple-mobillessfari
iCloud-com-apple-ncplugin-weather
iCloud-com-citrixon
iCloud-com-dayanand
iCloud-com-evernote
iCloud-com-fogcreek
iCloud-com-getdropt
iCloud-com-google-d
iCloud-com-kayak-tr
iCloud-com-microsof
iCloud-com-microsof
iCloud-com-microsof
iCloud-com-microsof
iCloud-com-microsof
iCloud-com-synology
iCloud-com-tinyspec
iCloud-com-waze-iphon
iCloud-com-zenlabs-c25k

```

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

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.



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Section 5: Part 4

Malware and Intrusion Analysis

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 82

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

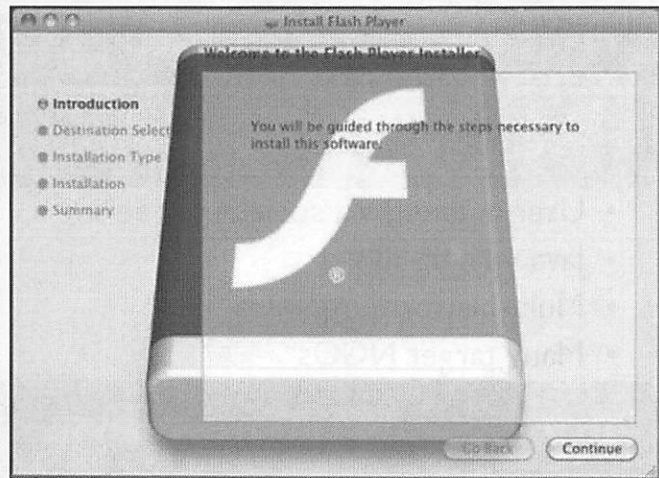


Image Source: <http://www.cultofmac.com/124840/new-flashback-os-x-trojan-is-in-the-wild-and-it-can-kill-os-xs-anti-malware-scams/>

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

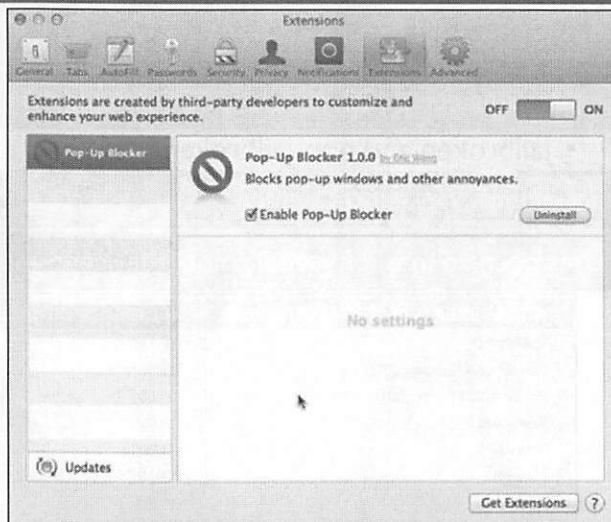


Image Source: <http://www.thesafemac.com/wp-content/uploads/2014/02/CoinThief-extension.png>

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

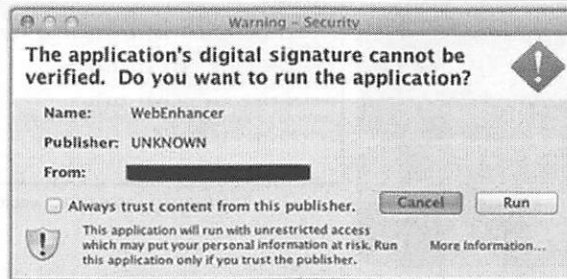
Rootkit and Spyware

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

```
joe -- bash -- 80x18
Joes-Mac-mini:~ joe$ codesign -dvvv macs.app/
Executable=/Users/joe/mac.app/Contents/MacOS/mac
Identifier=com.util.file
Format=bundle with Mach-O universal (i386 x86_64)
CodeDirectory v=20100 size=1362 flags=0x0(none) hashes=60+5 location=embedded
Hash type=sha1 size=20
CDHash=b0aa57a281c2d8cce6c9a09568c6e3fea52ff80e
Signature size=8514
Authority=Developer ID Application: Rajinder Kumar
Authority=Developer ID Certification Authority
Authority=Apple Root CA
Timestamp=Apr 8, 2013 11:52:49 AM
Info.plist entries=22
Sealed Resources rules=4 files=2
Internal requirements count=1 size=208
Joes-Mac-mini:~ joe$
```

<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 - Edit
Your computer has been locked and all your files has been encrypted with 2048-bit RSA encryption.
Instruction for decrypt:
1. Go to https://fiwf4kxysm4dpw5l.onion.to ( IF NOT WORKING JUST DOWNLOAD TOR BROWSER AND OPEN THIS LINK: http://fiwf4kxysm4dpw5l.onion )
2. use 1PGAU8qB1NcwSHYKnPhzCrfKyx8kxysmEoI as your ID for authentication
3. Pay 1 BTC (~487.47$) for decryption pack using bitcoins (wallet is your ID for authentication - 1PGAU8qB1NcwSHYKnPhzCrfKyx8kxysmEoI)
4. Download decrypt pack and run

--> Also at https://fiwf4kxysm4dpw5l.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/>

Intrusion Analysis

Autoruns

Internet
History

Email

Java Cache

Temp Files

Log
Analysis

Volume
Analysis

File
Quarantine

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.

Temp and Cache Directories /tmp, Java Temp and Cache

/tmp and /var/tmp

~/Library/Caches/Java/ or
~/Library/Application Support/Oracle/Java
Deployment/

- /Cache, /tmp directories
- IDX, JAR Files

Brian Baskin's (@bbaskin)

- Windows Executable
- Python Script

```
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: 55gP0Wmd1lIqDYd0F2EXCTPRpyU=
```

```
Name: mac  
SHA1-Digest: fvpEryer0UCRvrcUI0yvQTWj4Vs=
```

```
Name: win  
SHA1-Digest: f6fErX0tG88SsYClqc8kYTSFYIw=
```

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 93

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: 55gPOWmd1lIgDYd0F2EXCTPRpyU=
```

```
Name: mac
```

```
SHA1-Digest: fvpEryer0UCRvrcUI0yvQTwj4Vs=
```

```
Name: win
```

```
SHA1-Digest: f6fErX0tG88SsYClqc8kYTSFYIW=
```

Antivirus: File Quarantine

Introduced in 10.5

Quarantines downloaded files

Applications (Browsers, Email, IM, Airdrop)

Weaknesses

- Files on USB drives
- Applications that do not implement File Quarantine

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 [-] 5665726966696564446F776E6C6F6164506C7567696E2E7273726300}
    $b = {3C6B65793E4946506B67466C616744656661756C744C6F636174696F6E3C2F6865793E [-] 3C737472696E673E2F4C6962726
    172792F496E7465726E657420506C75672D496E732F3C2F737472696E673E}
    $c = {23212F62696E2F [0-2] 7368}
  condition:
    $a and $b and $c
}
```

Antivirus: Xprotect—XProtect.plist

The screenshot shows the XProtect.plist file with the following data:

Key	Value
Item 10	000 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 73 00 /tmp/launch-hs\
Item 11	015 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 73 65 /tmp/launch-hse
Item 12	030 00 2F 74 6D 70 2F 00 23 21 2F 62 69 6E 2F 73 \tmp/\#!/bin/s
Item 13	045 68 0A 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 h\mp/launch-h
Item 14	060 73 65 20 26 0A 6F 70 65 6E 20 2F 74 6D 70 2F se &lopen /tmp/
Item 15	075 66 69 6C 65 2E 64 6F 63 20 26 0A 0A 00 00 5F file.doc &f_\
Item 16	090 5F 50 41 47 45 5A 45 52 4F 00 00 5F 5F 6D 68 _PAGEZERO__mh
Item 17	105 5F 65 78 65 63 75 74 65 5F 68 65 61 64 65 72 _execute_header

Description	String	OSX.Mdropper.i
LaunchServices	Dictionary	(1 item)
LSItemContentType	String	com.microsoft.word.doc
Matches	Array	(1 item)
Item 0	Dictionary	(3 items)
MatchFile	Dictionary	(1 item)
MatchType	String	Match
Pattern	String	2F746D702F6C61756E63682D6873002F746D70
Item 18	Dictionary	(3 items)

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.

Key	Type	Value
▶ Item 10	String	000 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 73 00 /tmp/launch-hs\
▶ Item 11	String	015 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 73 65 /tmp/launch-hse
▶ Item 12	String	030 00 2F 74 6D 70 2F 00 23 21 2F 62 69 6E 2F 73 \tmp/\#!/bin/s
▶ Item 13	String	045 68 0A 2F 74 6D 70 2F 6C 61 75 6E 63 68 2D 68 h\tmp/launch-h
▶ Item 14	String	060 73 65 20 26 0A 6F 70 65 6E 20 2F 74 6D 70 2F se &fopen /tmp/
▶ Item 15	String	075 66 69 6C 65 2E 64 6F 63 20 26 0A 0A 00 00 5F file.doc &f\
▶ Item 16	String	090 5F 50 41 47 45 5A 45 52 4F 00 00 5F 5F 6D 68 _PAGEZERO\ _mh
▼ Item 17	String	105 5F 65 78 65 63 75 74 65 5F 68 65 61 64 67 72 _execute_header
Description	String	OSX.mdropper.i
▼ LaunchServices	Dictionary	(1 item)
LSItemContentType	String	com.microsoft.word.doc
▼ Matches	Array	(1 item)
▼ Item 0	Dictionary	(3 items)
▶ MatchFile	Dictionary	(1 item)
MatchType	String	Match
Pattern	String	2F746D702F6C61756E63682D6873002F746D70
▶ Item 18	Dictionary	(3 items)

Antivirus: GateKeeper

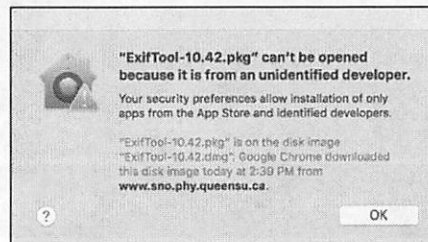
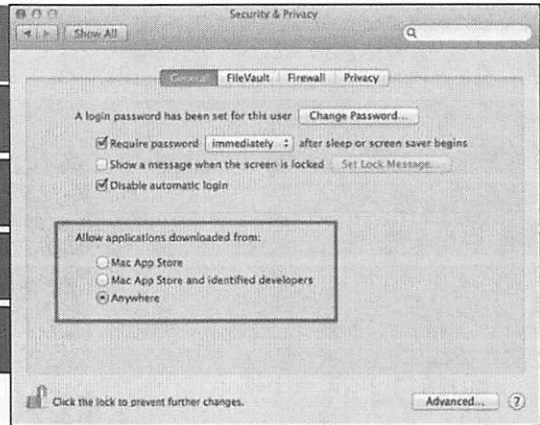
Introduced in 10.7.5

Antimalware Feature

Application Execution Restrictions

Security Settings

- Mac App Store
 - Users can only run apps from the store
- Mac App Store and identified developers
 - Default setting (10.8+)
 - Users can only run software signed using Apple Developer ID
- Anywhere
 - Default setting (10.7.5)
 - Users can run anything from anywhere
 - No longer available in 10.12



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.



Section 5: Part 5

Live Response

This page intentionally left blank.

Live Response: System Information

date

hostname

uname -a

sw_vers

```
bit:~ oompa$ date
Sun Aug 12 18:13:26 EDT 2012
bit:~ oompa$ hostname
bit
bit:~ oompa$ uname -a
Darwin bit 12.0.0 Darwin Kernel Version 12.0.0: Sun Jun 24 23:00:16 PDT 2012; ro
ot:xnu-2050.7.9~1/RELEASE_X86_64 x86_64
bit:~ oompa$ sw_vers
ProductName:    Mac OS X
ProductVersion: 10.8
BuildVersion:   12A269
```

SANS | **DFIR**

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 104

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_\(operating_system\)](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 *:*
```

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 (CLOSE_WAIT)	279	oompa	28u	IPv4	0x9ed2b1a710259ad5	0t0	TCP bit:61035->ec2-107-22-245-91.compute-1.amazonaws.com:https
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 UHLWiiir        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:~ ompa$ netstat -rn | more
Routing tables
Internet:
Destination          Gateway             Flags
192.168.1.254        192.168.1.254      UGSc
127.0.0.1           127.0.0.1         UCS
127.0.0.1           127.0.0.1         UH
169.254              link#6
169.254.204.125     b8:c7:5d:cc:5:80  UHLSW
192.168.1           link#6
192.168.1.1         c0:3f:e8:c:59:59  UHLWI
192.168.1.101      127.0.0.1         UHS
192.168.1.133      3c:7:54:3:65:20  UHLWI
192.168.1.209      d0:23:db:72:91:10 UHLWI
192.168.1.254      e0:69:95:50:4c:6  UHLWI
192.168.1.255      ff:ff:ff:ff:ff:ff UHLWI
Internet:
Destination          Use
192.168.1.254        1
192.168.1.209      45
192.168.1.133      4502
192.168.1.101      0
192.168.1.1         104
192.168.1           0
169.254              1
169.254.204.125     0
192.168.1           23616
192.168.1.1         0
127.0.0.1           0
127.0.0.1           0
192.168.1.254      22
Internet:
Destination          Rets
192.168.1.254        0
192.168.1.209      23
192.168.1.133      0
192.168.1.101      0
192.168.1.1         1
192.168.1           5
169.254              1
169.254.204.125     0
192.168.1           5
192.168.1.1         23616
192.168.1           0
127.0.0.1           0
127.0.0.1           0
192.168.1.254      22
Internet:
Destination          Netif
192.168.1.254        en1
192.168.1.209      en1
192.168.1.133      en1
192.168.1.101      lo0
192.168.1.1         en1
192.168.1           en1
169.254              en1
169.254.204.125     en1
192.168.1           lo0
192.168.1.1         lo0
192.168.1           en1
127.0.0.1           en1
127.0.0.1           en1
192.168.1.254      en1
Internet:
Destination          Expire
192.168.1.254        1
192.168.1.209      577
192.168.1.133      45
192.168.1.101      295
192.168.1.1         850
192.168.1           1188

```

Live Response:ARP Table

IPv4: `arp -an` or IPv6: `ndp -an`

```
bit:~ oompa$ arp -an
? (169.254.204.125) at b8:c7:5d:cc:5:80 on en1 [ethernet]
? (192.168.1.1) at c0:3f:e:8c:59:59 on en1 ifscope [ethernet]
? (192.168.1.133) at 3c:7:54:3:65:20 on en1 ifscope [ethernet]
? (192.168.1.209) at d0:23:db:72:91:10 on en1 ifscope [ethernet]
? (192.168.1.254) at e0:69:95:50:4c:6 on en1 ifscope [ethernet]
? (192.168.1.255) at ff:ff:ff:ff:ff:ff on en1 ifscope [ethernet]
```

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.

Live Response: Network Configuration

ifconfig

```
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,TS04>
    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 0xfffff00 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
```

SANS DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 112

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,TS04>
    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 0xffffffff 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 16876726 /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/Service
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 590016 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/Service
Google 8239 oompa txt REG 1,4 4202496 15374905 /Users/oompa/Library/Application Support/Google/Chrome/Default/Service
Google 8239 oompa txt REG 1,4 1056768 15374904 /Users/oompa/Library/Application Support/Google/Chrome/Default/Service
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$ ls -l | 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/Service
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/Service
Google 8239 oompa txt REG 1,4 4202496 15374905 /Users/oompa/Library/Application Support/Google/Chrome/Default/Service
Google 8239 oompa txt REG 1,4 1056768 15374904 /Users/oompa/Library/Application Support/Google/Chrome/Default/Service
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

```

Live Response: Users Logged On [1]

```
who -a
```

```
w
```

```
bit:~ oompa$ who -a
reboot  ~           Aug  4 11:24 00:24          1
oompa   console     Aug  4 11:26 old           57
oompa   ttys000     Aug 12 18:13 .           13949
oompa   ttys001     Aug 12 18:18 00:09       13976
.       run-level 3
bit:~ oompa$ w
18:55 up 8 days,  7:31, 3 users, load averages: 1.07 0.99 0.88
USER      TTY      FROM          LOGIN@  IDLE WHAT
oompa     console -              04Aug12 8days -
oompa     s000    -              18:13   - w
oompa     s001    -              18:18   9 /usr/bin/less -is
```

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 116

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/He
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.d
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/NetworkE
charlie   14673  0.0  0.2 2531632 16512  ?? S   7:10PM   0:00.48 /System/Library/CoreServices/Finder.d
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
```

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 118

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 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/Me
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.spdx
```

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



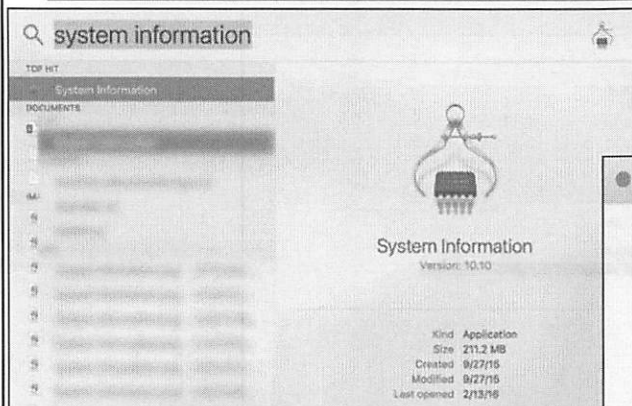
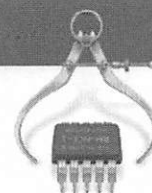
The `system_profiler` command is the command-line version of the System Information application.

The `system_profiler` command has different output formats and detailed level options. As long as the output is named with a “.spdx” 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
...
```

Live Response: System Information.app [1]

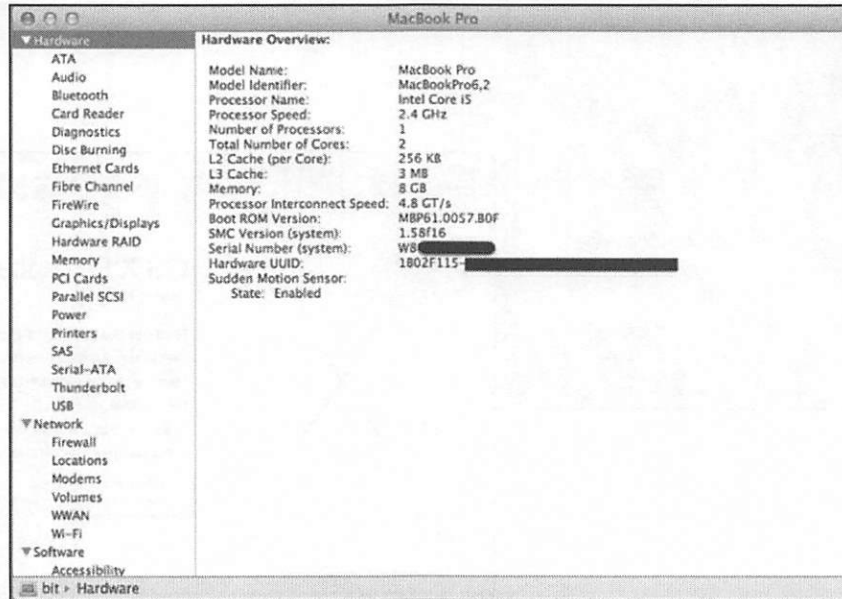


SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 121

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”.

Live Response: System Information.app [2]



SANS DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 122

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.

MacBook Pro

▼ Hardware

Hardware Overview:

Model Name:	MacBook Pro
Model Identifier:	MacBookPro6,2
Processor Name:	Intel Core i5
Processor Speed:	2.4 GHz
Number of Processors:	1
Total Number of Cores:	2
L2 Cache (per Core):	256 KB
L3 Cache:	3 MB
Memory:	8 GB
Processor Interconnect Speed:	4.8 GT/s
Boot ROM Version:	MBP61.0057.B0F
SMC Version (system):	1.58f16
Serial Number (system):	W8 [REDACTED]
Hardware UUID:	1B02F115-[REDACTED]
Sudden Motion Sensor:	
State:	Enabled

ATA

Audio

Bluetooth

Card Reader

Diagnostics

Disc Burning

Ethernet Cards

Fibre Channel

FireWire

Graphics/Displays

Hardware RAID

Memory

PCI Cards

Parallel SCSI

Power

Printers

SAS

Serial-ATA

Thunderbolt

USB

▼ Network

Firewall

Locations

Modems

Volumes

WWAN

Wi-Fi

▼ Software

Accessibility

bit ▶ Hardware

Lab 5.3

Malware and Live Response

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.



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Section 5: Part 6

Memory Acquisition and Analysis

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 126

This page intentionally left blank.

Memory Acquisition and Analysis Tools

Acquisition

OSXPMem

Volexity Surge

MacQuisition

Recon

Analysis

Volatility

Rekall

Volexity Volcano

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

GUI vs. CLI

RAM Image Output Types

- Raw (Padded or Unpadded)
- Mach-O
- E01
- ELF
- DMG

Admin Credentials Are Required!

- Installs a Kernel Extension

Compressed RAM

- 10.9+
- Broke Older De Facto Tools (Mac Memory Reader, Mac Memoryze)

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>

Memory Acquisition: OSXPMem

Created by Johannes Stuetzgen / Google / Rekall

Free and Open Source

Supports 64-bit 10.7+

AFF4 Map (default), Raw (padded), and ELF output formats

```
bash-3.2# ./osxpmem -o memory.dump
Imaging memory
Creating output AFF4 ZipFile.
Reading 0x8000 0MiB / 16250MiB 0MiB/s
Reading 0xbc8000 11MiB / 16250MiB 46MiB/s
Reading 0x18b8000 24MiB / 16250MiB 50MiB/s
Reading 0x2900000 41MiB / 16250MiB 64MiB/s
Reading 0x37a0000 55MiB / 16250MiB 58MiB/s
Reading 0x45e0000 69MiB / 16250MiB 56MiB/s
Reading 0x55c8000 85MiB / 16250MiB 62MiB/s
```

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 129

OSXPMem is a free and open-source tool created by Johannes Stuetzgen. 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>

Memory Acquisition: MacQuisition by BlackBag [1]

Created by BlackBag Technologies

Neither Free nor Open Source

Supports 10.6+

Raw, DMG, E01 (Uncompressed, Empty Block Compression, Fast Compression, Best Compression)

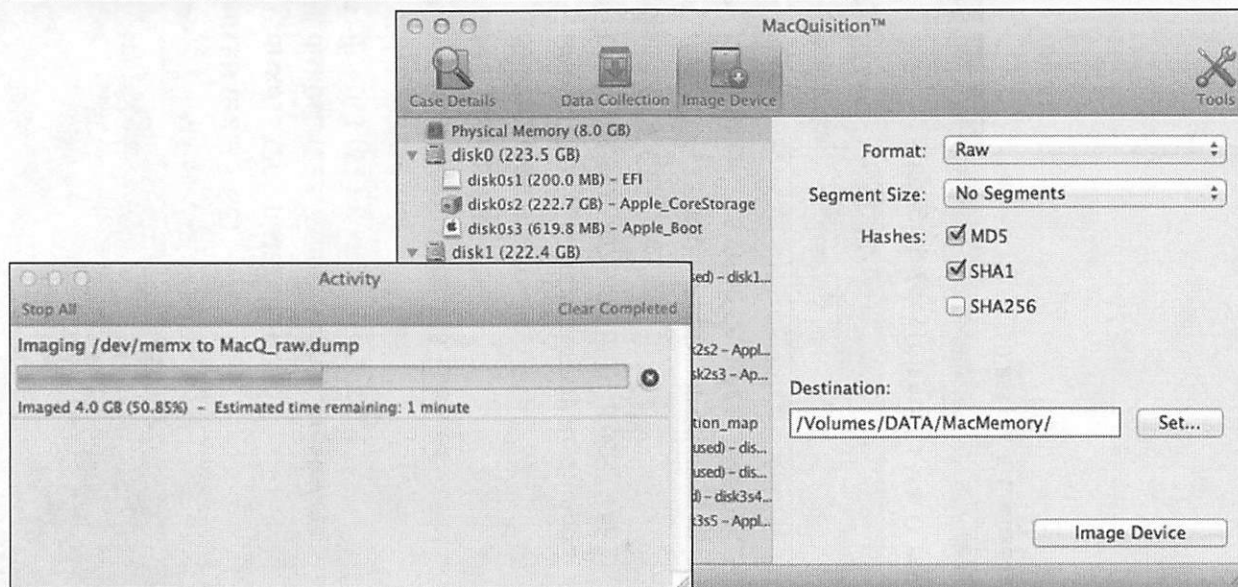
MacQuisition by BlackBag Technologies is their tool to capture all types of data in one shot. This tool allows an investigator to acquire incident response information, disk images, and RAM. While this tool makes it incredibly easy to capture everything you may need for an investigation, it does come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

Memory Acquisition: MacQuisition by BlackBag [2]

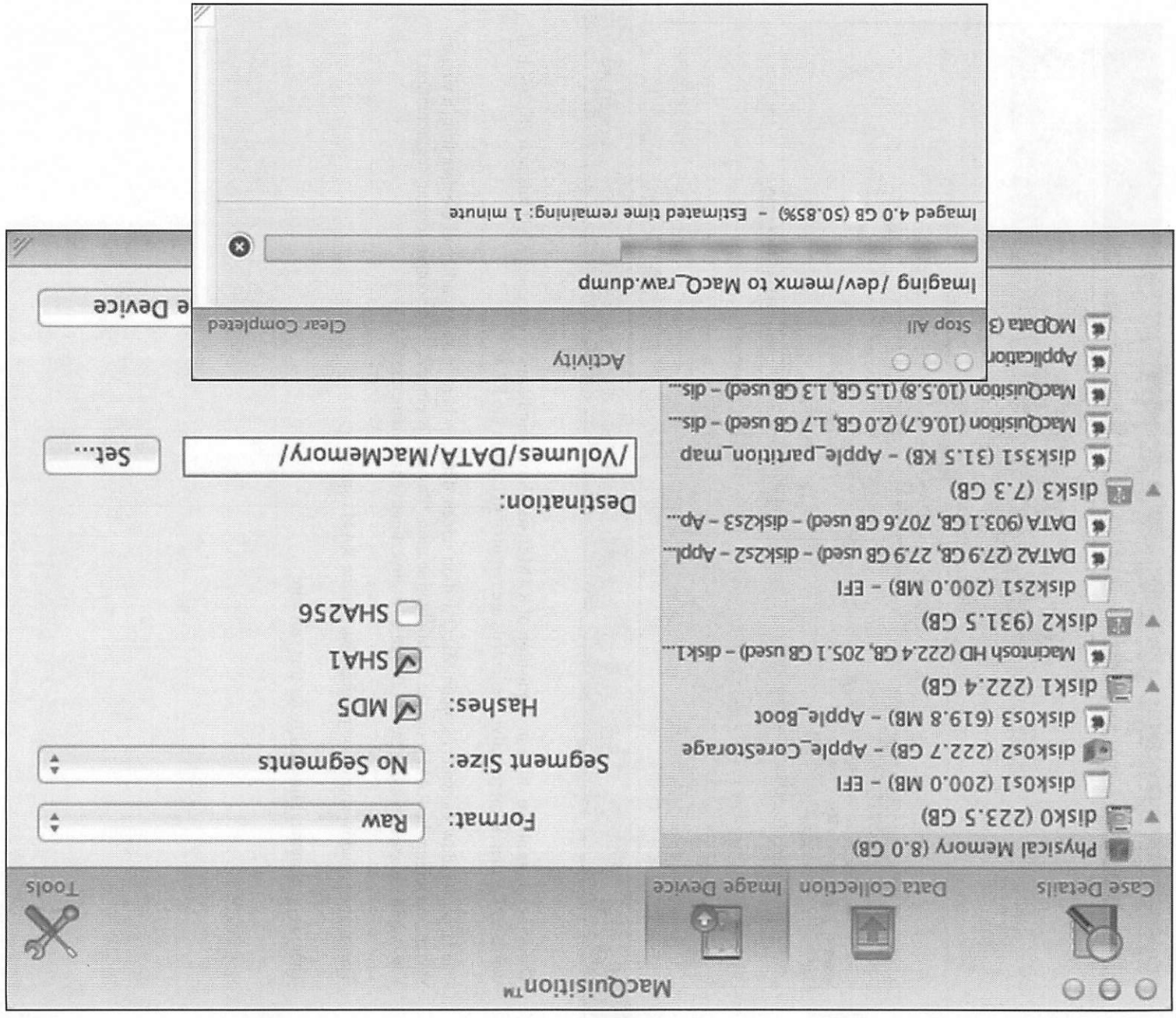


The screenshots above show an acquisition of RAM from a Mountain Lion system using MacQuisition 2012r3. It is worth noting that Administrative credentials are required to capture the RAM.

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.



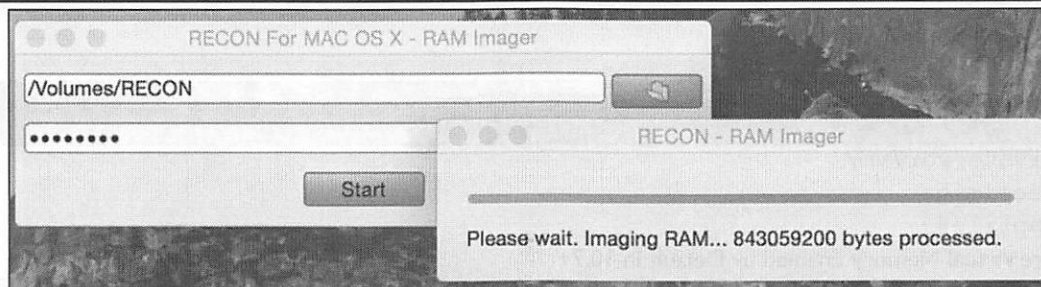
Memory Acquisition: Sumuri Recon

Created by Sumuri

Neither Free nor Open Source

Supports 10.7+

Raw Output Format



SANS | **DFIR**

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 133

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

Benefits and Limitations

Same basic functionality

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

Sleep/Swap Files (10.6-)

- /private/var/vm/
- sleepimage
- swapfile#
- Secure Virtual Memory Enabled by Default in 10.7+

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.

Memory Analysis Tools: Volatility vs. Rekal

Volatility

Official Mac support in Volatility 2.3

Python-based

Plugins

CLI

Supports images from 10.5+

- Mach-O,
- Raw (Padded)
- DMG
- VMEM

github.com/volatilityfoundation/volatility

Rekal

Forked Volatility to be more modular

Python-based

Plugins

CLI or web console GUI

Supports images from 10.6+

- RAW
- ELF
- Mach-O
- VMEM

github.com/google/rekal

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.

Rekal 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.

Rekal 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.rekal-forensic.com/>

<https://github.com/google/rekal>

Memory Analysis Profiles and Plugins (Volatility vs. Rekall)

Volatility	Rekall
<ul style="list-style-type: none">• Profiles<ul style="list-style-type: none">• Only comes with Windows profiles. Must download macOS profiles or build your own.• https://github.com/volatilityfoundation/profiles• <code>python vol.py --profile <profilename></code>• Plugins<ul style="list-style-type: none">• Expected basic plugins• More supported plugins<ul style="list-style-type: none">• Malware, keychain, file dumps	<ul style="list-style-type: none">• Profiles<ul style="list-style-type: none">• Auto finds and downloads from GitHub server the correct profile. Must be connected to the internet.• Can download profiles for offline usage. https://github.com/google/rekall-profiles• <code>rekall --profile <profile></code>• Plugins<ul style="list-style-type: none">• Expected basic plugins

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, next 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
Mach0AddressSpace - Address space for mach-o files to support atc-ny memory reader
mac_version - Prints the Mac version
machoinfo - Dump Mach-0 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>

Memory Analysis: Volatility Usage

```
python vol.py --profile=<profile> -f <memory image> <plugin>
```

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

```
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_version
Volatility Foundation Volatility Framework 2.4
Darwin Kernel Version 13.1.0: Thu Jan 16 19:40:37 PST 2014; root:xnu-2422.90.20~2/RELEASE_X86_64
```

SANS | **DFIR**

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 138

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
launched	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 `launched` (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 `launched`. In the screenshot example, the `Terminal` application has multiple `login` processes. Each of these `login` processes, in turn, has a child `bash` process.

Memory Analysis: Volatility—Processes

```
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
0xffffffff80337a74a8 fontworker          5844 501 20 5844 64BIT  0x4b6c000 2015-03-15 21:26:54 UTC+0000
0xffffffff803581a5d8 timezoned           5832 210 210 5832 64BIT  0x7c4b3000 2015-03-15 21:26:34 UTC+0000
0xffffffff8036ad0000 AirPort Base Sta 5813 501 20 5813 64BIT  0x55e63000 2015-03-15 21:26:29 UTC+0000
0xffffffff80350e0df8 rpcsvchost          5806 0 0 5806 64BIT  0x74fe7000 2015-03-15 21:26:29 UTC+0000
0xffffffff80337aa7e0 netbiosd            5805 222 222 5805 64BIT  0x47561000 2015-03-15 21:26:29 UTC+0000
0xffffffff8035817748 digest-service      5804 0 0 5804 64BIT  0x7d588000 2015-03-15 21:26:29 UTC+0000
0xffffffff8036ad3338 syncdefaultsd       5682 501 20 5682 64BIT  0x6d5c4000 2015-03-15 21:26:29 UTC+0000
0xffffffff803b662bf0 mdworker            5667 501 20 5667 64BIT  0x3bd41000 2015-03-15 21:26:29 UTC+0000
```

mac_pslist

```
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
```

mac_pstree

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 140

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
0xffffffff80337a74a8 fontworker         5844 501  20  5844 64BIT  0x4b6c000 2015-03-15 21:26:54 UTC+0000
0xffffffff803581a5d8 timezoned          5832 210 210  5832 64BIT  0x7c4b3000 2015-03-15 21:26:34 UTC+0000
0xffffffff8036ad0000 AirPort Base Sta  5813 501  20  5813 64BIT  0x55e63000 2015-03-15 21:26:29 UTC+0000
0xffffffff80350e0df8 rpcsvchost         5806  0    0  5806 64BIT  0x74fe7000 2015-03-15 21:26:29 UTC+0000
0xffffffff80337aa7e0 netbiosd           5805 222 222  5805 64BIT  0x47561000 2015-03-15 21:26:29 UTC+0000
0xffffffff8035817748 digest-service     5804  0    0  5804 64BIT  0x7d588000 2015-03-15 21:26:29 UTC+0000
0xffffffff8036ad3338 syncdefaultsd      5682 501  20  5682 64BIT  0x6d5c4000 2015-03-15 21:26:22 UTC+0000
0xffffffff803b662bf0 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

```


Memory Analysis: Volatility—Network

```
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

mac_ifconfig

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	74.125.130.108	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	74.125.130.108	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

mac_netstat

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 142

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.

Section 5: Part 7

Password Cracking and Encrypted Containers

This page intentionally left blank.

Password Cracking and Encrypted Containers

User Passwords

Keychains

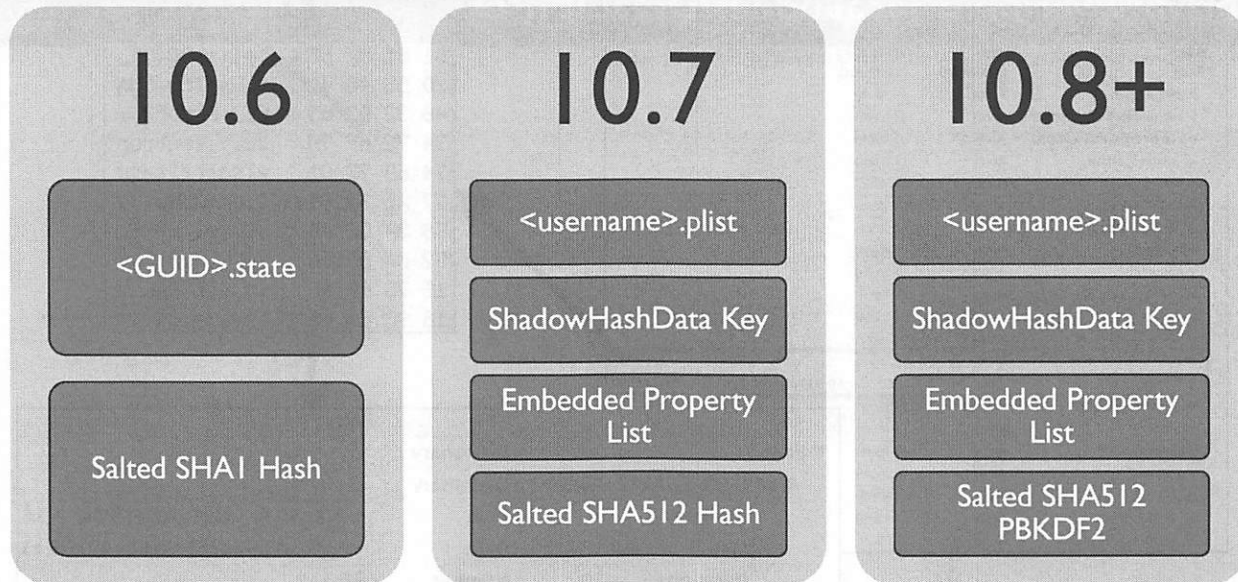
FileVault 2

Encrypted DMGs

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.

Password Cracking: Historical Password Shadows



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!)

Password Shadow 10.8+ /private/var/db/dslocal/nodes/Default/users

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)
▶ passwd	Array	(1 item)
▶ gid	Array	(1 item)
▶ _writers_hint	Array	(1 item)
▶ _writers_jpegphoto	Array	(1 item)

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

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 D7 03 512-PBKDF2..
74 72 6F 70 .....Wentrop
74 65 72 61 yTsaltZitera
77 AE 33 64 tions0..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

Password Cracking Software

John the Ripper

- Support for 10.6 and 10.7, 10.8+ hashes
- Jumbo build compiled specifically for macOS
- Free

Hashcat

- Support for 10.6, 10.7, and 10.8+ hashes (v.46)
- Free

Passware

- Support for 10.6, 10.7, and 10.8+ hashes
- Not Free

DaveGrohl

- Support for 10.6 (v1.0), 10.7, and 10.8+ hashes (v2.1)
- Distributed
- Free

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>

Accessing FileVault Volumes: FileVault 2

User Password

- `hdiutil attach -readonly -nomount -stdinpass filevault2image.dmg`

Master Password

- `security unlock-keychain FileVaultMaster.keychain`
- `diskutil corestorage unlockvolume <UUID> -recoverykeychain FileVaultMaster.keychain`

Recovery Key

- `diskutil corestorage unlockvolume <UUID> -passphrase <recovery key>`
- Recovery key stored with Apple: Contact Apple subpoenas@apple.com

Brute Force Tools

- Passware, PRTK, HashCat

iCloud Access

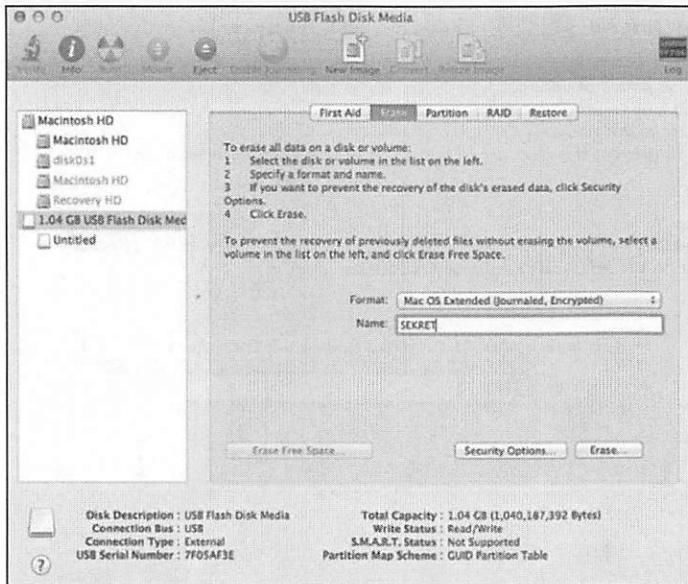
- Elcomsoft EPB w/Apple ID and password

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/>

Encrypted Non-OS FileVault Volumes



Are you sure you want to erase the disk "USB Flash Disk Media" and create an encrypted partition?

Erasing a partition deletes all the data on that partition but does not affect other partitions on the same disk.

By setting a password, the partition will be encrypted and not accessible without the password.

WARNING: Files on this partition will be encrypted using this password. If you forget the password, your data will be lost.

New password:

Verify:

Password Strength: Weak

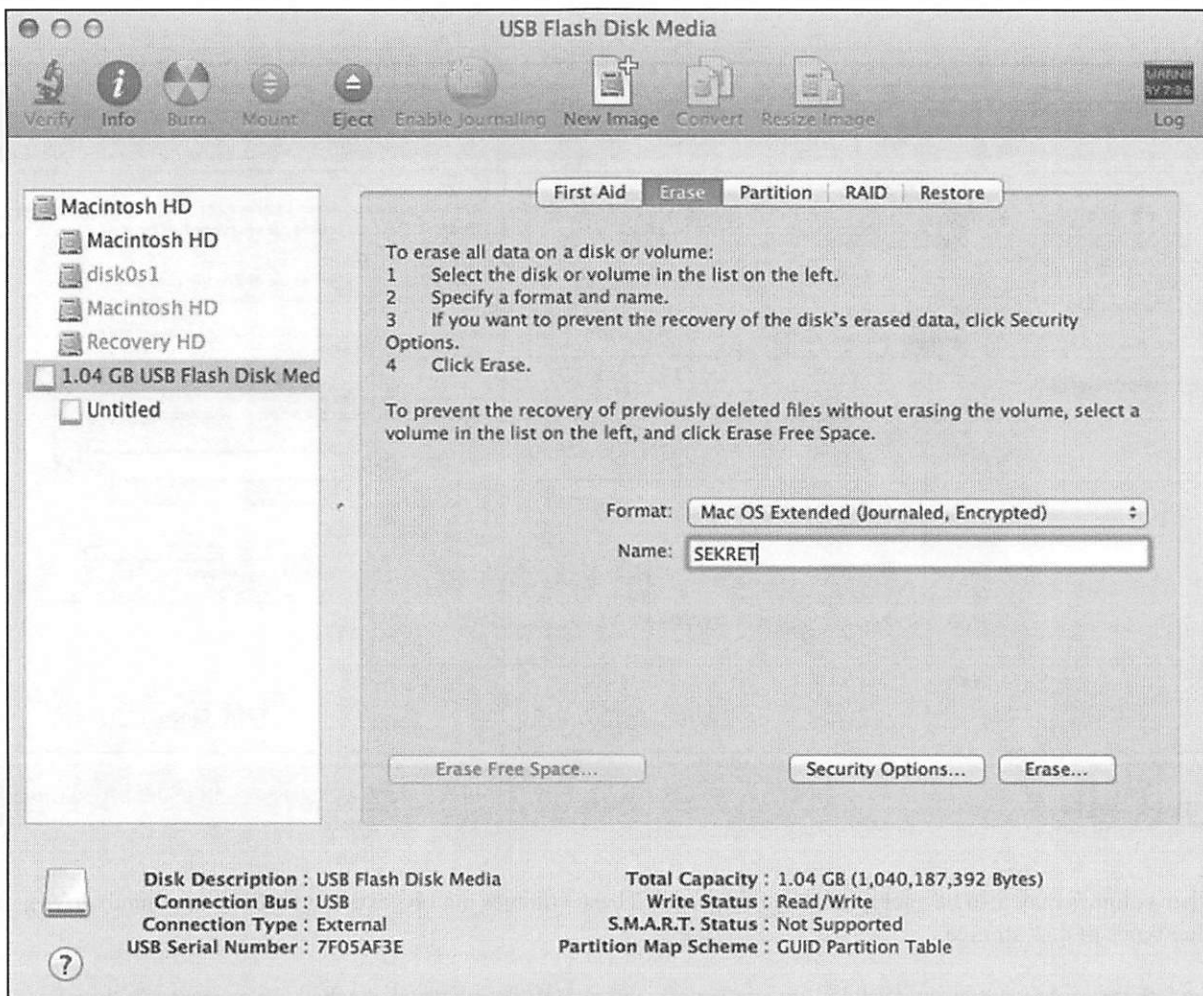
Hint:

Cancel Erase

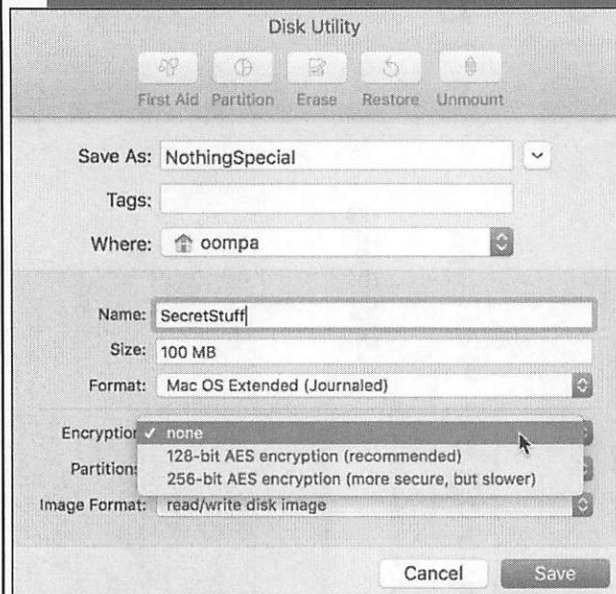
Other volumes may also be encrypted using FileVault. These volumes may be external hard drives, thumb drives, or other types of disk storage.

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.



Encrypted Disk Images (DMGs)



0x00000100 = 256

```

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 B8 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
    
```

0x00000080 = 128

```

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 ....}][...0...
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
    
```

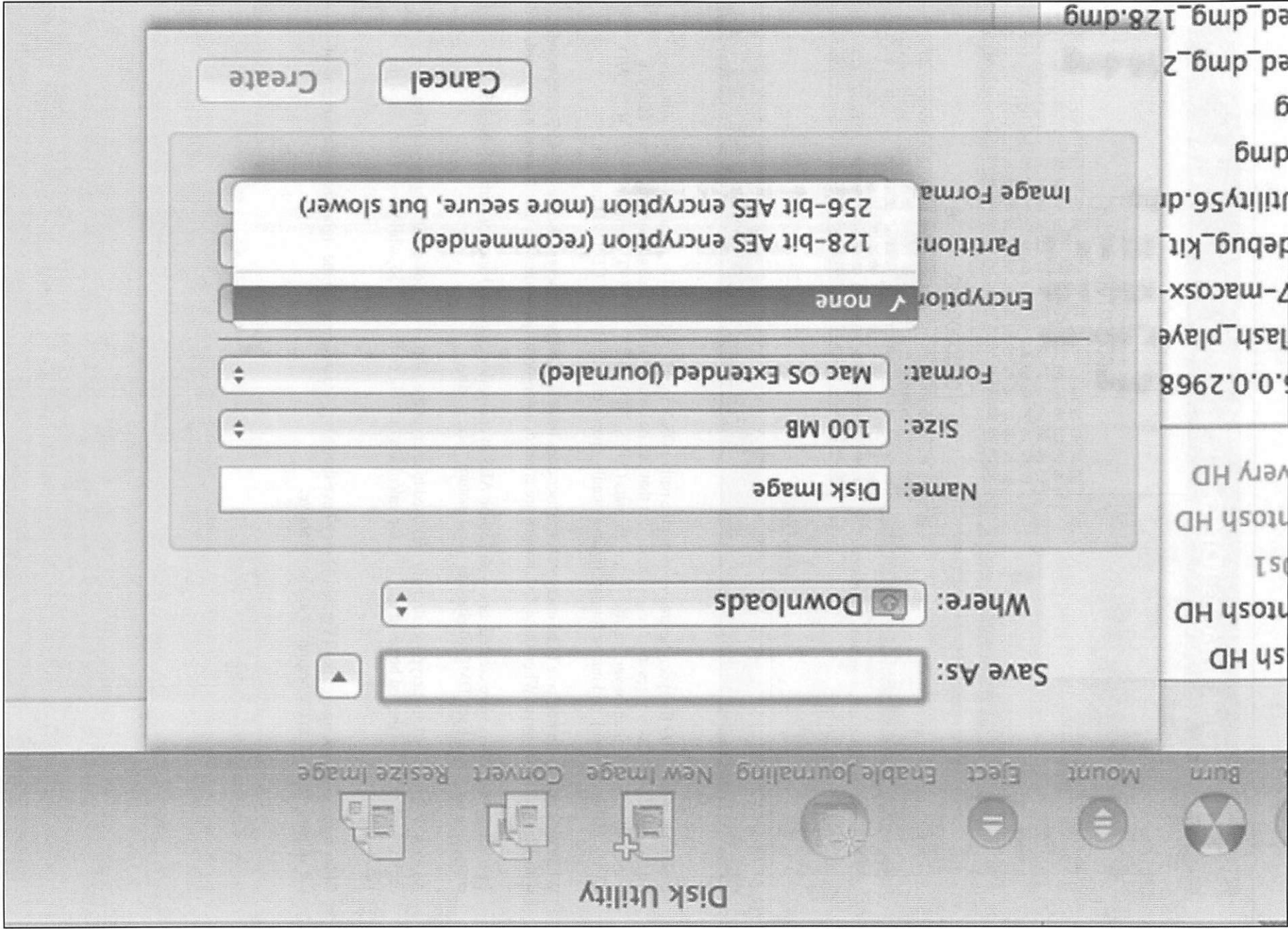
Disk Images (or DMG files) can be created using the native OS X tool Disk Utility or via the command line using `hdiutil`. These disk images are encapsulated files that can be created in any size and can contain any data you want. Other options include various file system formats (APFS, HFS+, FAT, ExFAT), 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.



Cracking and Accessing Encrypted Volumes

```
hdiutil attach -readonly -nomount -stdinpass  
sekretstuff_USB.dmg
```

John the Ripper

- Extract hash using `dmg2john` (Available in Jumbo release)

CrowbarDMG

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.

Creating a Password Cracking Dictionary File

May make brute forcing password faster.

There may be plaintext passwords in the memory image

Additional filters can be made with command-line utilities (strings/awk/etc.) to filter string list down by characters and string size

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

```
./john --wordlist=dictionary.txt user_password_hash.txt
```

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
```



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Lab 5.4 – Memory Analysis, Password Cracking, and Encrypted Containers

SANS DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 160

This page intentionally left blank.



FOR518 Section 5: Advanced Analysis Topics

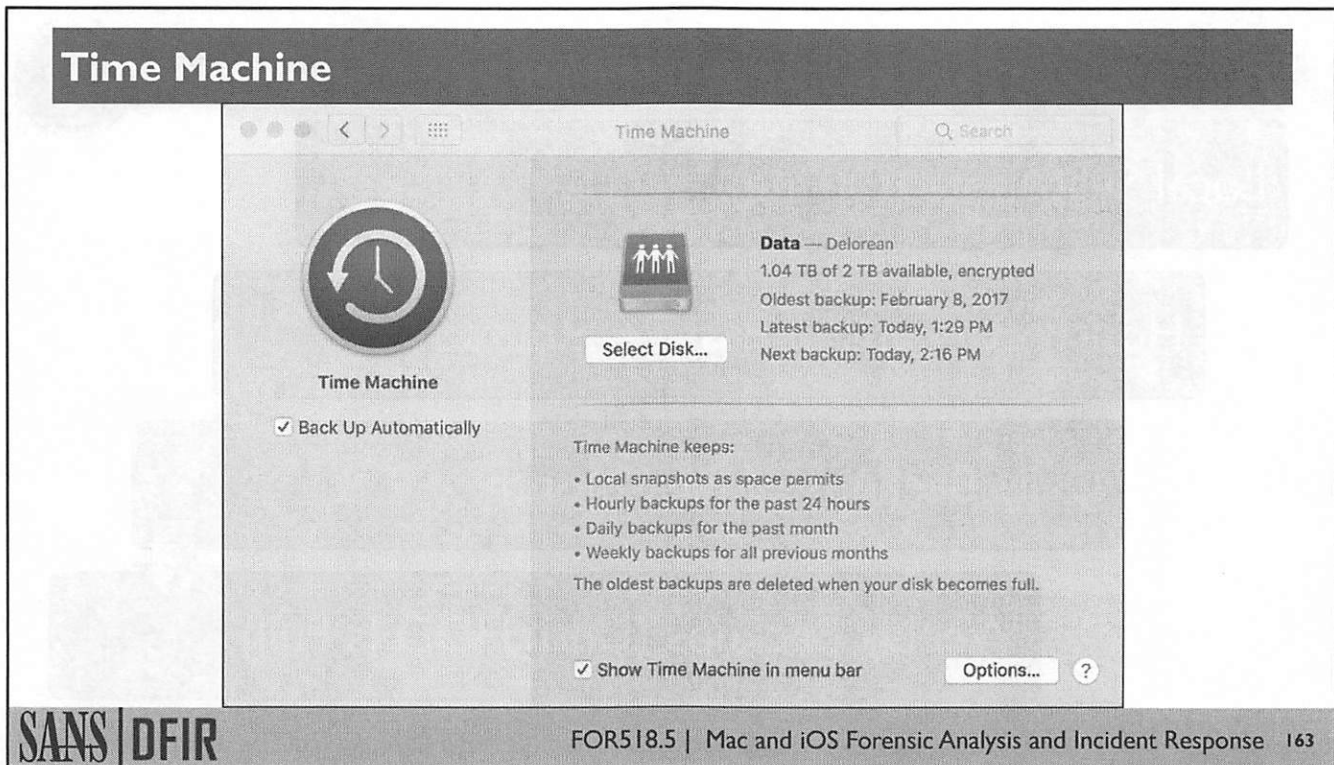
© 2020 Sarah Edwards | All Rights Reserved | Version F01_01

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: Backup Schedule and Storage



Local Snapshots (Offline)

Hourly Backups of Past 24 Hours

Daily Backups of Past Month

Weekly Backups (Until HDD Is Full)

SANS | DFIR

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 164

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.backup.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>

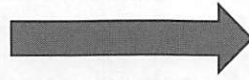
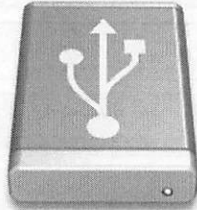
Mountain Lion: Power Nap

<https://support.apple.com/en-us/HT204015>

Time Machine: Local vs. Network Disks



External Hard Drive



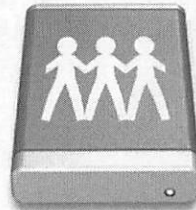
Local



Time Capsule/NAS



Network



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.

Time Machine: Terminology

Backup Source	<ul style="list-style-type: none">• Volume to be backed up
Backup Disk	<ul style="list-style-type: none">• Volume containing backups
Backup Destination	<ul style="list-style-type: none">• Local Destination: Synonym for Backup Disk• Network Destination: AFP Share where backups reside
Backup Disk Image	<ul style="list-style-type: none">• Sparsebundle containing backups
Backup Store	<ul style="list-style-type: none">• “Backups.backupdb” directory
Machine Directory	<ul style="list-style-type: none">• Directory containing backups for one system• “Dade’s Mac”
Snapshot	<ul style="list-style-type: none">• Directory inside Machine Directory containing backup files
Snapshot Volume	<ul style="list-style-type: none">• Directory inside snapshot of volume backed up

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

Reference:
tmutil Man Page

Time Machine Preferences /Library/Preferences/com.apple.TimeMachine.plist

Root	Dictionary (10 items)	
LastDestinationID	String	852420F8-06F2-4C94-8B20-D2EFC0155A92
LastCompactTime	Number	1,516,892,074
HostUIDs	Array (1 item)	
Item 0	String	502A32AD-3CAF-5685-DC09-053E285901C8
LocalizedDiskImageVolumeName	String	Time Machine Backups
BackupAlias	Data	-D000000003C80002 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	-D000000004100002 00010444 61746100 00000000
TimeCapsuleName	String	DeLorean
ConsistencyScanDate	Date	Mar 9, 2018 at 6:38:01 AM
DestinationUIDs	Array (2 items)	
Item 0	String	AEB9C8DF-F13C-39EC-986D-1163A90B2138
Item 1	String	F567E2C0-64B2-31A4-BD86-674310C2B2E8
ReferenceLocalSnapshotDate	Date	Mar 11, 2018 at 1:30:19 PM
RESULT	Number	10
LastKnownEncryptionDate	String	Encrypted
DateOfLatestWarning	Date	Mar 26, 2018 at 8:43:28 PM
DestinationID	String	552420F8-06F2-4C94-8B20-D2EFC0155A92
BytesUsed	Number	1,357,859,663,872
FirmwareCheckDate	Date	Mar 6, 2018 at 11:27:20 PM
SnapshotDates	Array (7 items)	
Item 0	Date	Mar 9, 2018 at 9:38:01 AM
Item 1	Date	Mar 9, 2018 at 11:47:26 PM
Item 2	Date	Mar 10, 2018 at 12:48:30 PM
Item 3	Date	Mar 11, 2018 at 11:04:11 AM
Item 25	Date	Mar 11, 2018 at 12:24:45 PM
Item 26	Date	Mar 11, 2018 at 1:33:22 PM
RootVolumeUUID	String	1D19162C-619C-3A34-A02C-2D428A48C44E
BytesAvailable	Number	640,047,771,648
SkipPaths	Array (1 item)	
Item 0	String	/Users/Shared/jad
AutoBackup	Boolean	NO

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 3	Date	Mar 11, 2018 at 11:30:19 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/adj
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 00
027 00 00 00 00 00 00 00 00 00 00 00 CA AE 2D 3E 48 2B 00 01 00 00 00 00 01 04 44 61 74
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
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
108 00 00 00 00 00 00 00 00 00 02 CA AE 2D 3E 00 00 00 00 00 00 00 00 00 FF FF FF FF
135 00 31 11 61 73 00 00 00 00 00 00 00 00 00 00 00 04 44 61 74 61 00 10 00 00
162 00 00 CA AE 65 7E 00 00 00 11 00 08 00 00 CA AE 65 7E 00 00 01 00 00 00 00 02 00
189 09 44 61 74 61 3A 44 61 74 61 00 00 00 0E 00 0A 00 04 00 44 61 00 74 00 61 00 0F
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
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
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
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
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
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
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
405 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
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
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
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
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
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
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
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
621 2E 5F 74 63 70 2E 6C 6F 63 61 6C 2E 08 02 0A 08 0C 84 02 24 00 00 00 00 00 00 00
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
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
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
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
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
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
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
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
864 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
891 45 64 77 61 72 64 73 00 00 18 00 3B 61 66 70 3A 2F 2F 53 61 72 61 68 25 32 30
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
945 70 2E 5F 74 63 70 2E 6C 6F 63 61 6C 2E 2F 44 61 74 61 00 FF FF 00 00
.....Data.....
.....->H+.....Dat
a.....
.....->.....
1.as.....Data...
...e~.....e~
>Data;Data.....D.a.t.a.
.....D.a.t.a...../Volume
s/Data.....afpm.....&F
.j.....a.7.W.....u.....
.....Delorean.....
.....Data.....
.....Sarah E
dwards.....
....."Delorean_afpovertcp
_tcp.local.....$.
.....
.....Sarah
Edwards.....;afp://Sarah%20
Edwards@Delorean_afpovertc
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	/.fsevents
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.

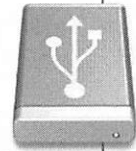
▼ Root	Dictionary	(4 items)
▼ PathsExcluded	Array	(25 items)
Item 0	String	/.MobileBackups
Item 1	String	/MobileBackups.trash
Item 2	String	/.MobileBackups.trash
Item 3	String	/.Spotlight-V100
Item 4	String	/.TemporaryItems
Item 5	String	/.Trashes
Item 6	String	/.com.apple.backupd.mvlist.plist
Item 7	String	/.fsevents
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.

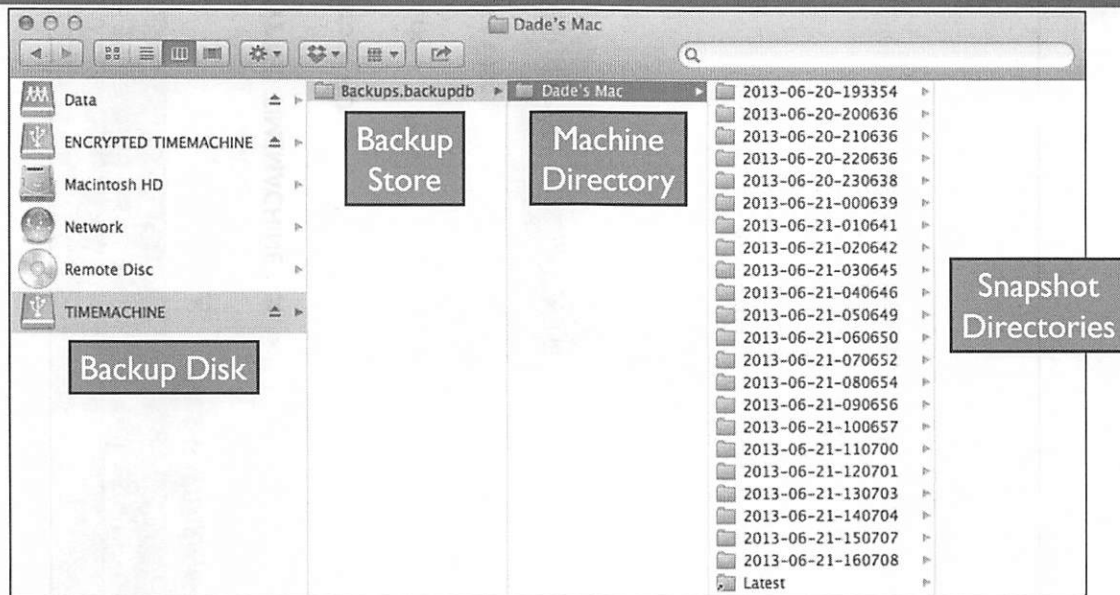
The screenshot also shows us the volume UUID, disk size, and connection protocol (USB).

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

Time Machine: Structure—External Backup Disk [2]

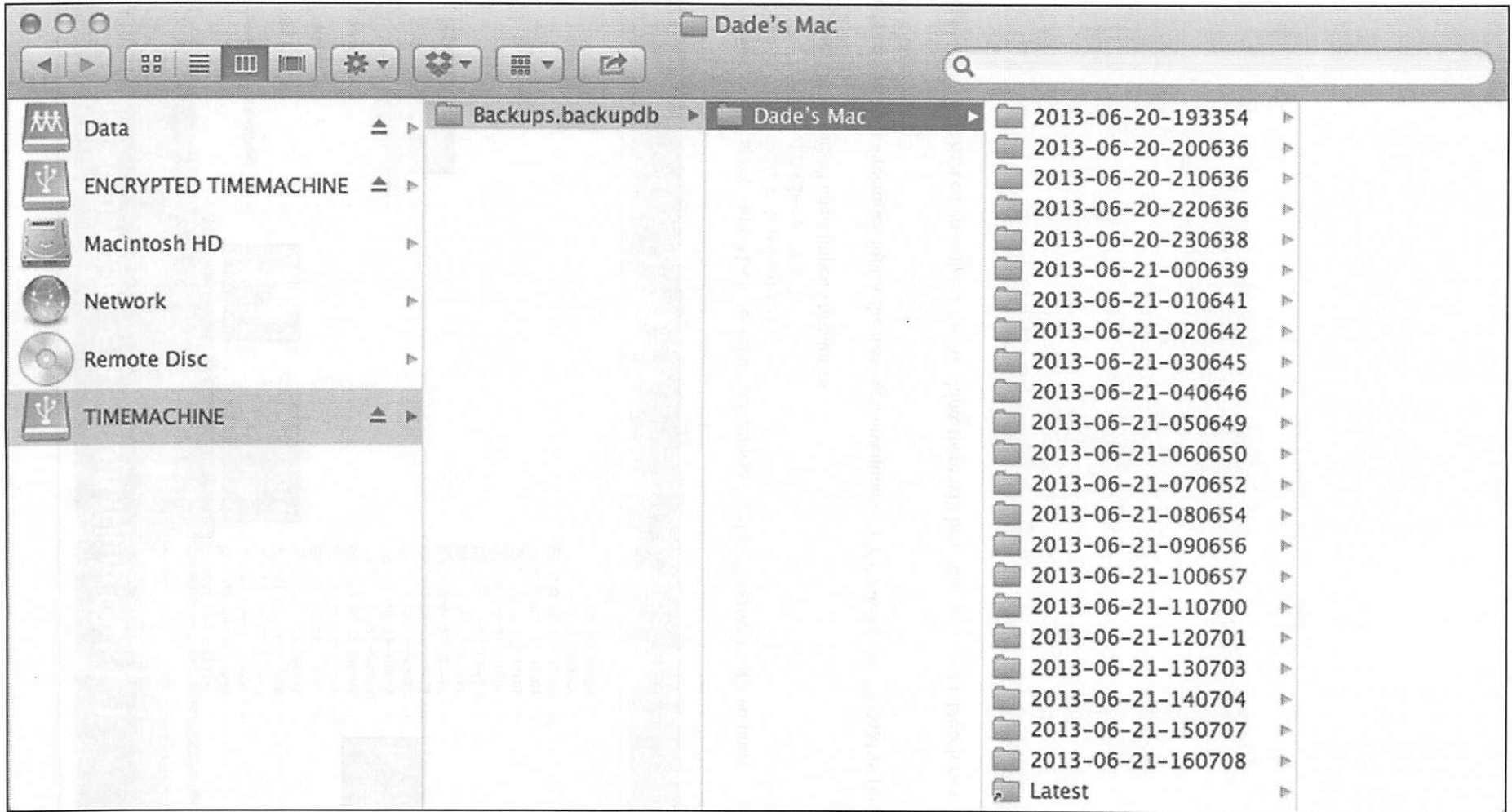


Using Finder, we can look at the Time Machine–related directories on the TIMEMACHINE volume:

- 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.



Time Machine [1] Structure of Encrypted External Backup Volume

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: A9908865-8EF5-3CD2-81CE-9EF960857C09

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-B535-31468687A885



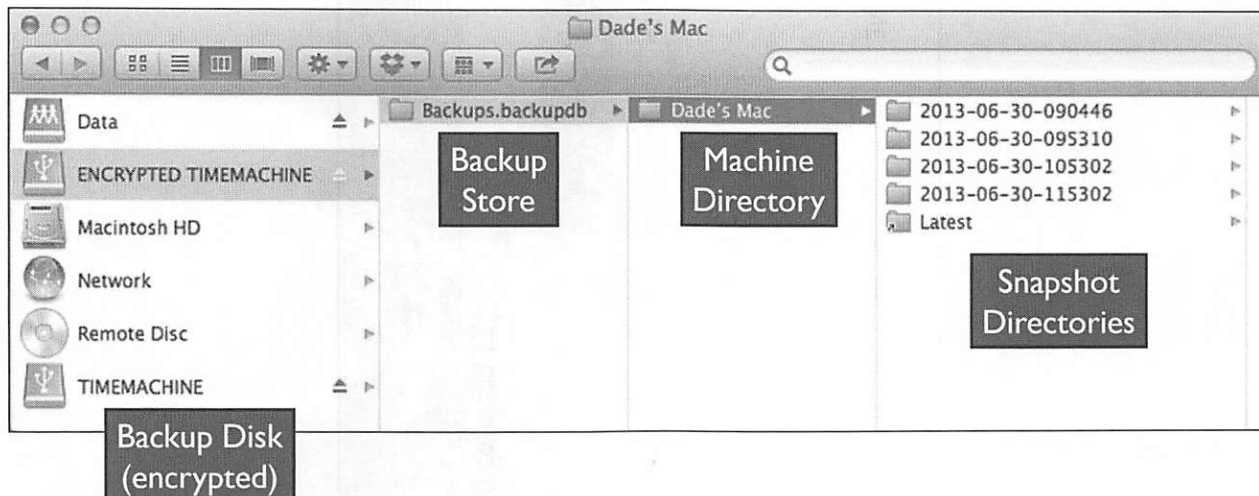
When an encrypted Time Machine is mounted (for example, from a forensic image), it will ask for a password, as shown in the screenshot above. This password allows access to mount the encrypted CoreStorage volume as shown in the larger screenshot above.

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

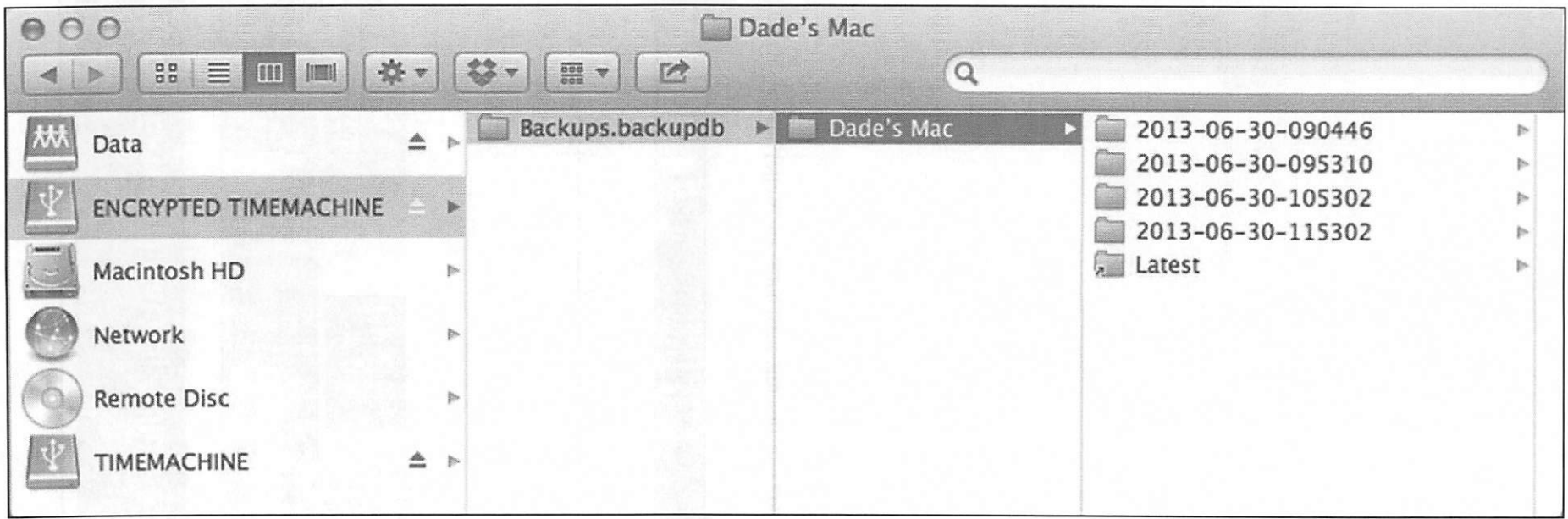
ENCRYPTED TIMEMACHINE:

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-B535-314B8687A885

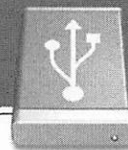
Time Machine [2] Structure of Encrypted External Backup Volume



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
```

SANS | **DFIR**

FOR518.5 | Mac and iOS Forensic Analysis and Incident Response 179

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.backup.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.backup.HasRecoverySet:
00000000 59 45 53 |YES|
00000003
com.apple.backup.HostUUID:
00000000 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 |00000000-0000-10|
00000010 30 30 2D 38 30 30 30 30 30 30 30 43 32 39 39 39 |00-8000-000C2999|
00000020 36 42 38 45 00 |6B8E.|
00000025
com.apple.backup.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@ 1 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.

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

Time Machine: External Disk Structure Snapshots Extended Attributes

```
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
```

Snapshot Type:
1: Monthly
2: Hourly
3: Daily



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 /.fsevents: 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 04 00 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 |23E685CA-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
```

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 |23E685CA-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/Dade's Mac/2013-06-20-193354/Macintosh HD
nibble:Macintosh HD sledwards$ ls -la
total 16304

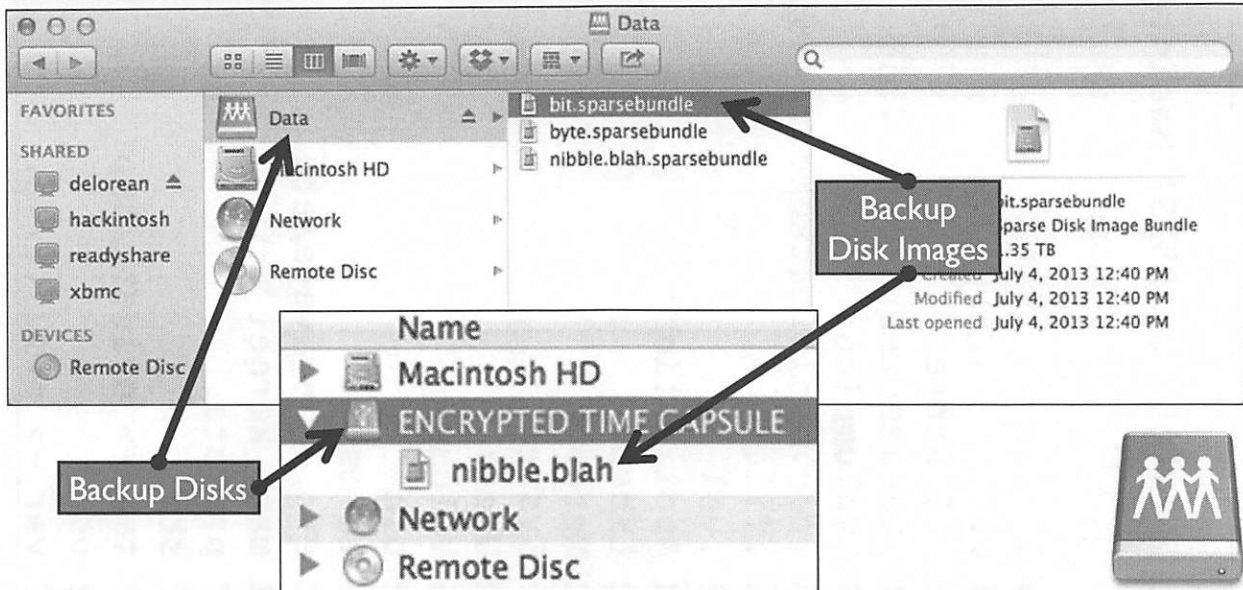
```

```

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
drwxr-xr-x@ 34 root admin 1156 May 18 21:08 Applications
drwxr-xr-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
drwxr-xrwt@ 2 root admin 68 Jun 20 22:05 Volumes
drwxr-xr-x@ 39 root wheel 1326 Sep 23 2012 bin
drwxr-xr-t@ 2 root admin 68 Jun 20 2012 cores
drwxr-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
drwxr-xr-x@ 1 root wheel 11 Sep 23 2012 tmp -> private/tmp
drwxr-xr-x@ 10 root wheel 340 Sep 23 2012 usr
drwxr-xr-x@ 1 root wheel 11 Sep 23 2012 var -> private/var

```

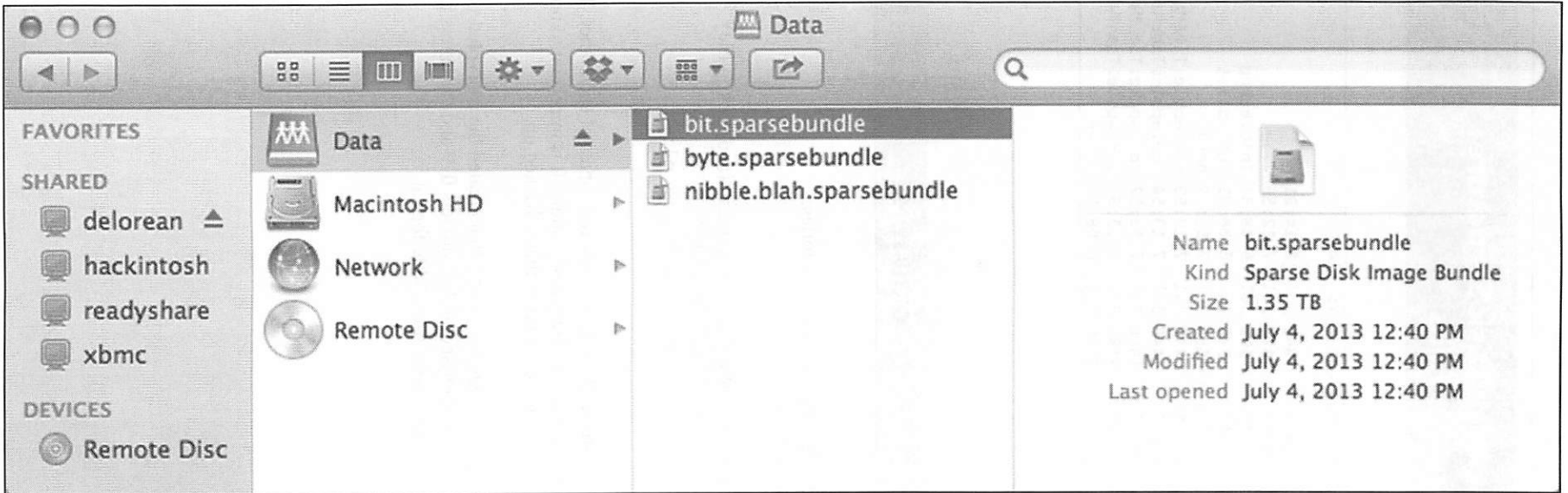

Time Machine: Structure of Network Backup Disk



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

Time Machine: Local Snapshots

Offline Backups

Introduced in 10.7

Locations:

- `/.MobileBackups`
- `/Volumes/MobileBackups` (MTMFS)

`com.apple.TimeMachine.plist`

- MobileBackups Key (YES = Enabled)

Similar Extended Attributes

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”.

Time Machine: Local Snapshots – /Volumes/MobileBackups

```
nibble:Volumes sledwards$ tree -L 5 MobileBackups/
```

```
MobileBackups/  
├── Backups.backupdb  
│   └── nibble.blah  
│       ├── 2013-07-08-092057  
│       │   └── Macintosh\ HD  
│       │       ├── Applications  
│       │       ├── Groups  
│       │       ├── Library  
│       │       ├── Network  
│       │       ├── Shared\ Items  
│       │       ├── System  
│       │       ├── User\ Information -> /Library/Docu  
│       │       ├── Users  
│       │       ├── Volumes  
│       │       ├── bin  
│       │       ├── cores  
│       │       ├── etc -> private/etc  
│       │       ├── extracted  
│       │       ├── mach_kernel  
│       │       ├── opt  
│       │       ├── private  
│       │       ├── sbin  
│       │       ├── tmp -> private/tmp  
│       │       ├── usr  
│       │       ├── var -> private/var  
│       │       └── Latest -> 2013-07-08-092057
```

Volume Name	Type	Mount Point
home	autofs	/home
MobileBackups	mtmfs	/Volumes/MobileBackups
net	autofs	/net

MobileBackups:

Type: mtmfs
Mount Point: /Volumes/MobileBackups
Mounted From: localhost:/giG2DZ6kEZN2_H16Woskw3
Automounted: No

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.

Time Machine Mounting from a Forensic Image

Network Backup Volume and Encrypted Network Volumes

- `hdiutil attach timemachine.sparsebundle -readonly`

External Backup Volume

- `hdiutil attach timemachine.dmg -readonly`

External and Encrypted Backup Volume

- `hdiutil attach timemachine.dmg -nomount -readonly`

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 unquiesze

```
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
152.7K /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
```

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 `unquiesze` 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$ tutil 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

Time Machine: `tmutil compare`

Perform a “diff” on:

- Two snapshots
- Snapshot and live system

Comparisons:

- Extended attributes
- ACLs
- File sizes and modes
- UIDs and GIDs
- Modification timestamp
- Data fork

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
! 1810 (mtime) /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/.com.apple.backup.nvlist.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_00000000-0000-1000-0000-000C2959608E.plist
! 100.0K (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 (size, mtime) /Volumes/TIMEMACHINE/Backups.backupdb/Dade's Mac/2013-06-20-200636/Macintosh HD/private/var/db/.TimeMachine.Results.plist
! 2020 (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: 00
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$ tutil 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_00000000-0000-1000-8000-000C2999588E.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
```


Time Machine: `tmutil compare` (XML Output)

▼ 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

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



SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Bonus Lab 5.5 – Time Machine

This page intentionally left blank.

Index

.emlx	4:78-79
.GlobalPreferences.plist	2:9-10
.jar	5:93
/cache	3:81, 4:11, 4:13, 4:25, 4:44, 4:49-50, 4:54, 4:62-63, 5:20-24, 5:26-27, 5:93, 5:186
/data	1:88, 2:46, 3:87, 4:13, 4:16-17, 4:44, 4:46, 4:49, 4:55, 4:59, 4:61-62, 4:72, 4:81, 4:131, 4:145, 4:162-163, 5:71-72, 5:75
/download	1:16, 2:67, 2:77, 2:133, 2:135, 2:152-171, 3:17, 3:169, 4:65, 4:81, 5:150
/etc/fstab	1:45, 1:52
oxAF	1:103, 1:105
oxCAFEBABE	3:74
oxCEFAEDFE	3:74
oxCFFAEDFE	3:74
oxEE	1:103
oxFEEDFACE	3:74
oxFEEDFACF	3:74
25pp	1:52
802.11	2:15, 3:179

A

Access Date	2:45, 2:194, 2:196
Access Point (AP)	2:13, 2:18, 2:20-21, 2:30, 3:11, 3:176-177, 3:179, 5:26, 5:28, 5:74
AccessData	5:120
account number	4:8
Acquisition	1:21, 1:23, 1:31-35, 1:43, 1:48, 1:54-55, 1:57-60, 1:79-80, 2:11-13, 3:13, 3:48, 3:87, 3:111-112, 4:11, 4:18, 4:20, 4:49, 4:71, 5:17, 5:24, 5:126-131, 5:133, 5:159
Acquisition and Analysis	1:33, 1:58, 5:126-127
Active Directory (AD)	2:24, 3:158-159
Address Book	1:49, 4:19, 4:23, 4:118-121, 4:123, 4:125
Address Resolution Protocol (ARP)	5:111
administrative privileges	2:23
administrator	1:38, 2:23, 3:168-169, 5:128
Advanced Encryption Standard (AES)	1:111, 5:155

Advanced Persistent Threat (APT)	5:91
AES	1:111, 5:155
AIM	2:135, 3:16
AirDrop	2:77, 2:108, 2:120-121, 5:97
Algorithm	5:147-148, 5:150
alias	2:45, 3:23, 3:41, 3:44, 3:58, 4:65, 5:97, 5:167
aliases	2:45, 3:23, 3:41, 3:44, 3:58, 4:65, 5:97, 5:167
AMD	5:138
American Standard Code for Information Interchange (ASCII)	4:170
Anti-Malware	5:84
Anti-Virus	5:95-99, 5:101
Antivirus	5:95-99, 5:101
Apple Developer ID	5:88, 5:101
Apple File Conduit Protocol (AFC)	1:54
Apple File Connection (AFC)	1:54
Apple Filing Protocol (AFP)	5:167
Apple Pay	4:147-151, 4:153, 4:179
Apple Remote Desktop (ARD)	3:61
Apple Tech Note 1150	2:140, 2:142, 2:175, 2:181-184, 2:187, 2:189, 2:191-194, 2:200, 2:202
Apple Watch	1:10, 3:70, 3:118, 4:177-181, 5:9
Application Level Firewall (ALF)	3:57-58
APT28	5:91
Arbitration	1:32
artifacts	1:30, 1:62, 2:13, 2:77, 2:125-127, 3:28, 3:139, 3:180, 5:201
ASCII	3:55, 4:170
Attribute Modification Date	2:194, 2:196
audit_class	3:122-123
audit_control	3:122-123
Auditing	3:17, 3:119
auditreduce	3:127
Authentication	1:48-49, 3:70, 3:124-125, 5:70, 5:112
Authenticator	3:14
Authorization	2:40, 3:169, 4:26-27
AutoRun	3:17-19, 3:21-26, 5:92

B

B-tree	2:129, 2:154, 2:159, 2:161-171, 2:181-184, 2:186-187, 2:189, 2:191, 2:200-201
Backdoor	5:87-88
Backup Date	1:68, 2:174, 2:194, 2:196
balanced tree	2:181
Base64	1:20, 1:73
bash	3:7-9, 4:35, 5:139
Basic Security Module (BSM)	3:98, 3:119, 3:127
Beacon	3:179
Big Endian	1:101, 2:140, 2:174-175, 2:184-186, 2:189, 2:196, 2:204, 5:155
Bill of Materials (BOM)	3:84-85
binaries	1:30, 1:38-39, 1:45, 3:74, 5:93, 5:135
binary	1:15-20, 1:40, 1:73, 2:24, 2:38, 2:44, 2:50- 51, 2:53, 2:55, 2:57-58, 2:60, 2:70, 2:74, 2:81, 2:142, 2:181, 2:204, 3:23, 3:55, 3:74, 3:102, 3:110, 3:115, 3:119, 3:150, 4:45, 4:67, 4:82, 4:89, 4:93, 4:120, 5:38, 5:40, 5:93, 5:167, 5:188
BIOS	2:17, 3:73
bit	1:54, 2:21, 2:46, 2:55, 2:121, 3:28, 3:73-74, 3:141, 3:148-149, 3:170, 4:131, 5:129, 5:151, 5:155
bit shifting	3:63
BlackBag Technologies	1:12, 1:21, 2:86, 5:130
BlackLight	1:21, 1:41, 1:79, 2:50, 2:89, 2:122, 2:141, 2:151, 4:12, 5:45
Block	1:40, 1:83, 1:92, 1:117, 2:14, 2:141-142, 2:147, 2:152, 2:154-160, 2:170, 2:174-175, 2:186, 2:191, 2:197, 2:201, 2:204, 3:58, 5:186-187, 5:197
Blocks	1:40, 1:83, 1:92, 1:117, 2:147, 2:152, 2:155- 160, 2:170, 2:174-175, 2:186, 2:197, 3:58, 5:186-187
Bluetooth	3:67, 3:69-70, 4:21, 4:178, 4:180, 5:120
Bookmarks.db	4:45
Bookmarks.plist	2:46, 4:45
BOOT_TIME	3:154
Brute-force	1:74, 1:76-77, 4:8, 5:151-152, 5:158
Brute-Force Attack	1:76-77
BSD	2:168, 2:194, 3:107, 3:119

Bydia 1:52
 Bypass 1:31, 1:59
 bz2 3:101
 bzcat 3:101

C

cache 1:27, 2:7, 2:112, 2:114, 3:69, 3:81, 3:148,
 4:10-11, 4:13-14, 4:25, 4:44, 4:49-50,
 4:52, 4:54, 4:61-63, 4:72, 4:112, 4:114,
 5:20-27, 5:92-93, 5:186
 Cache.db 4:11, 4:14, 4:49-50, 4:52
 calculatedrift 5:207
 CalDAV 4:103-105, 4:109
 Calendar 1:48, 1:61, 2:33-34, 2:36, 2:107, 4:19,
 4:21-23, 4:101-105, 4:107, 4:109, 4:111-
 112, 4:114, 4:118, 4:179, 5:57
 CallHistory.storedata 4:96
 Carrier 1:50, 1:103, 2:12, 2:30, 2:168, 2:197, 3:89,
 4:99
 carve 1:14
 Carving 1:14, 2:86, 3:56
 Casper 1:109
 Casper Suite 1:109
 Catalog File 2:141, 2:174, 2:181-182, 2:189, 2:191-194,
 2:200
 Cauliflowervest 1:109
 Celebrite 5:64
 checkout4mac 3:17
 checksum 2:72, 2:154, 3:85, 5:93
 chunk 5:34, 5:45, 5:47-48, 5:50, 5:52
 Class Key 1:48, 2:55, 2:158
 CoinThief 5:85
 Collection 3:111, 5:26
 com.apple.account.google 3:16
 com.apple.ipod.plist 1:62
 Command History 3:7-8
 Command-and-Control (C2) 5:85
 Communication 2:36, 2:108, 3:25, 3:69, 4:86-87, 5:64
 Compression 1:117, 2:68, 2:78, 3:100-101
 ConfigurationModificationDate 2:110
 connections 2:21, 3:58, 3:144, 3:176, 5:12, 5:105, 5:107,

	5:109, 5:114
Consent	4:20-21
Console.app	3:102, 3:111-112, 3:116
consolidated.db	4:25
Container	1:27-28, 1:87, 1:99, 1:113, 2:46, 2:145-146, 2:153-160, 2:170, 3:46, 3:87, 3:91, 4:8, 4:13, 4:16, 4:44, 4:46, 4:49, 4:55, 4:59, 4:61-62, 4:81, 4:131, 4:145, 5:8, 5:14, 5:71- 72, 5:75, 5:145-146, 5:160
Content Modification Date	2:194, 2:196
context	1:32, 1:116, 2:51, 2:100, 2:102, 2:131, 3:8, 3:16, 3:25, 3:43, 3:100-101, 3:153, 4:36, 4:38
Conversations	4:16, 4:89
Conversion	1:21, 4:36
Cookie	4:14, 4:45, 4:67
Correlation	3:130-131, 3:181
Counter	1:12, 1:98, 2:168
Creation Date	2:8, 2:174, 2:194, 4:133
Crisis	5:87, 5:93
Crisis/Morcut	5:87
CrowbarDMG	5:158
Cryptograph	1:46, 4:132
csh	1:1, 1:120, 2:1, 2:121, 3:1, 3:182, 4:1, 4:183, 5:1, 5:161
curl	2:119
Cyclic Redundancy Check (CRC)	1:105, 3:85
Cyclical Redundancy Check (CRC)	1:105, 3:85
Cydia	1:52-53, 3:18

D

Data Correlation	3:130-131, 3:181
Data Layer	1:83
Data Protection	1:46, 1:48-49, 3:14
Data Records	2:182, 2:189, 2:201
DCIM	4:165-166
dd	1:33, 1:117, 2:68, 2:78, 2:171
DeCode	2:29, 2:68, 4:175
decoding	2:197, 4:174-175
Decryption	2:160
default permissions	1:24

deleted 1:50, 2:23, 2:27-28, 2:86, 2:114, 2:131-132, 3:93, 3:167, 4:19, 4:77, 4:79, 4:99, 4:133, 4:137, 5:188
 Detection 5:136
 DeviceCache 3:69
 DeviceLock 2:14
 DeviceRegistry 4:180-181
 DeviceRegistry.state 4:180
 DHCP 2:17-18, 5:186
 dictionaries 1:74, 1:77
 dictionary 1:17, 1:76-77, 2:57, 2:107, 3:23, 3:33-34, 5:13, 5:159
 Dictionary Attack 1:76-77
 Diff 5:209
 Digital Camera Images (DCIM) 4:165-166
 diskutil 1:36-37, 1:40, 1:85, 1:87-90, 1:92, 1:94, 1:99, 1:110-111, 1:113, 2:146, 2:149, 2:155, 2:160
 dmg2john 5:158
 Domain Name System (DNS) 1:27-28, 2:49, 3:19, 3:21, 3:72, 3:89, 3:106, 3:115, 4:7, 5:36, 5:79
 Drag and Drop (DnD) 1:74, 3:170
 Dropbox 2:165, 5:67, 5:107
 Dynamic Host Configuration Protocol (DHCP) 2:17-18
 Dynamic-text.dat 3:33

E

Effaceable Storage 1:46, 1:48
 EFI 1:32-33, 1:58, 1:83, 1:85, 1:87, 1:90, 1:100, 1:103, 1:105, 1:107, 1:109, 2:17, 3:64, 3:66, 3:73, 3:119, 4:65, 5:19, 5:48, 5:64, 5:134
 EFI Protective MBR 1:103
 Elcomsoft 1:53, 1:74, 3:14, 5:64, 5:66-67, 5:70, 5:152
 Elcomsoft Phone Breaker 1:74, 3:14, 5:64, 5:66-67, 5:70
 EnCase 2:68, 5:128
 encryption 1:31, 1:33, 1:35, 1:44-46, 1:48-49, 1:54, 1:61, 1:64, 1:109, 1:111, 1:113, 1:116, 2:158, 2:160, 2:170, 5:152, 5:155
 escrow 1:59, 1:109
 Evernote 3:170

Evidence	1:32, 1:52, 2:89, 3:172-173
Ewfmount	1:39
Executable and Linkable Format (ELF)	5:128-129
EXFAT	1:116, 2:127-129, 3:39, 3:77, 5:155
ExifTool	2:66, 2:97, 4:160
Exploits	1:50
export	2:100, 3:98
EXT	4:166
Extents Overflow File	2:141, 2:174-175, 2:181-182, 2:191, 2:200
extraction	1:21, 1:55, 2:34, 2:191, 4:71, 4:74, 5:64

F

Facetime	3:16, 4:88, 4:96
FaceTime	3:16, 4:88, 4:96
Failed Login Count	2:26
Failed Login Timestamp	2:26
FAT file system	2:128, 3:77
File Key	1:48, 2:192
File Record	2:193-196, 2:199-200
File System Acquisition	1:79, 2:11
File System Extraction	1:55, 2:34, 4:74
File System Layers	1:83
File System Metadata	2:108, 2:147, 2:161
File System Structure	2:152
File Thread Record	2:193
FileVault	1:31, 1:36, 1:40, 1:85, 1:87-88, 1:100, 1:109-111, 1:113, 1:116, 2:23, 2:29, 2:146, 5:146, 5:152-153
Firewall	3:57-58, 5:120
FireWire	1:31, 3:37, 5:112, 5:120
Flash	1:31, 3:85, 3:170, 5:84, 5:87
Flashback	3:170, 5:84
Folder Record	2:189, 2:193-194
Folder Thread Record	2:193
Forensic Analysis	1:3, 1:97, 1:102-103, 2:88, 2:168, 2:197, 3:35, 3:93, 5:128
Forensics	1:1, 1:30, 1:60, 1:97, 1:120, 2:1, 2:110, 2:201, 3:1, 3:182, 4:1, 4:67, 4:91, 4:183, 5:1, 5:64, 5:161
Frequent Locations	5:18, 5:24
fsck	2:72, 3:146

FSEvents	2:86, 2:88-89, 2:91, 2:93, 2:95, 2:97, 2:99-100, 2:102, 2:126-127, 3:135, 5:186, 5:188
fsstat	2:178
fstab	1:45, 1:52
FTP	1:58, 5:186
Full Disk Encryption (FDE)	1:31, 1:111
Fusion Drive	1:111, 1:115
FXDesktopVolumePositions	3:39

G

Gatekeeper	5:101
GeoHistory.mapsdata	4:170-172
geolocation	4:163, 4:171, 4:173
Globally Unique ID (GUID)	1:36, 1:40, 1:59, 1:68, 1:83-85, 1:97-101, 1:105, 1:107, 2:32, 2:36, 2:47, 2:60, 2:86, 2:141, 3:9-10, 3:12, 3:87, 4:13, 4:44, 4:46, 4:49-50, 4:52, 4:55, 4:59, 4:61-62, 4:65, 4:76-79, 4:103, 4:119-121, 4:123, 4:125, 4:131, 4:141-142, 4:145, 4:171-173, 4:180- 181, 5:8, 5:14, 5:36, 5:43, 5:75, 5:97, 5:147, 5:155
GPS	4:25, 4:160, 4:178, 5:24, 5:26
GPT Header	1:97-99, 1:105
GPT.dmg	2:141, 2:178, 2:195
graphical user interface (GUI)	1:9, 1:11-12, 1:50, 1:52, 2:19, 2:30, 2:91, 2:109, 3:18, 3:23-24, 3:28, 3:36-37, 3:57, 3:163, 4:28, 4:31, 4:103, 5:116, 5:122, 5:128, 5:130, 5:133, 5:135
grep	1:32, 3:160, 5:114, 5:136-137
Guest Account	2:29
GUID Partition Tables (GPT)	1:85, 1:87, 1:89-90, 1:97-103, 1:105, 1:107, 1:110, 2:141, 2:178, 2:195
gzcat	3:101
Gzip	2:86, 2:100, 3:100-101, 4:132, 4:137, 4:140
gzip	2:86, 2:100, 3:100-101, 4:132, 4:137, 4:140

H

hard disk drive (HDD)	1:115, 2:86, 3:158, 3:173, 5:167
Hash	1:32, 2:30, 3:66, 4:54, 5:147-148, 5:151, 5:158
Hashcat	1:74, 5:150, 5:152
hashes	4:54
Hashing	5:147, 5:150
hdiutil	1:36, 1:38-40, 1:116-117, 2:178, 5:155, 5:158, 5:204
Heap	1:57-58
hex	1:17, 1:19, 1:21, 1:62, 2:21, 2:47, 2:71, 2:74, 2:81, 2:95, 2:154, 4:47, 4:93, 4:140, 4:170, 5:26, 5:28, 5:36, 5:93
Hex editor	1:17, 2:47, 4:170
HFS+	1:12, 1:38-40, 1:44, 1:85, 1:90, 1:94, 1:99-100, 1:103, 1:110, 1:116-117, 2:68, 2:71, 2:78, 2:82, 2:86, 2:91, 2:126, 2:128, 2:140-142, 2:152, 2:161, 2:167-169, 2:173-175, 2:178, 2:181, 2:184-186, 2:189, 2:191, 2:194, 2:196, 2:204, 3:39, 3:139, 3:146, 5:155, 5:171
hiberfil.sys	5:134
Hibernation	5:134
History.plist	3:83, 4:46, 5:75, 5:194, 5:199
Hosts File	3:66
Hotspots	2:21, 4:25
HyperText Transfer Protocol (HTTP)	4:11

I

iBackupbot	1:16
iCal	4:102-105, 4:107, 4:109, 4:111, 4:118
icat	2:78, 2:82, 2:191
iCloud	1:46, 1:61, 1:64, 2:29, 2:121, 3:10-14, 3:16, 4:17, 4:22, 4:38, 4:46-47, 4:72, 4:102, 4:105, 4:118-119, 4:130, 4:132, 4:139, 4:149, 4:157, 4:159, 4:169, 5:36, 5:40, 5:56-61, 5:63-64, 5:66-67, 5:69-71, 5:74, 5:78-79, 5:152
iCloud Drive	5:64, 5:66-67, 5:70, 5:78
iCloud keychain	3:10-14

IconState.plist	3:30
Identification	1:70, 1:97, 2:36, 3:106
identifier	1:62, 1:68, 1:83, 1:85, 1:94, 1:100, 2:36, 2:60, 3:72, 3:87, 3:89, 3:93, 3:106, 3:124, 3:141, 3:143, 4:33, 4:55, 4:59, 4:65, 4:135, 4:137, 4:143, 4:151, 5:40, 5:57, 5:79, 5:97, 5:179
iExplorer	1:79, 4:24
ifconfig	5:112, 5:142
iLoot	5:69
Image Type	5:204
imageinfo	1:117
imagent	3:16
IMAP	2:34
iMessage	2:121, 4:88, 4:90
import	1:15, 1:28, 2:108, 3:11, 3:14, 3:36, 3:112, 3:133, 3:135, 3:167, 4:10, 4:25, 4:31, 4:33, 5:112, 5:188, 5:194
Incident Response (IR)	1:3, 3:8, 5:130
Index Nodes	2:182
Indicator	3:150
Info.plist	1:68, 2:11, 3:26, 3:48, 3:71-72, 3:77, 4:103-105, 5:95, 5:194
Inline Attribute Data Record	2:201-202
Inode	1:73, 2:78, 2:82, 2:114, 2:147, 2:163-171, 2:191, 2:194, 3:49-50, 5:41
Installation	1:32, 1:83-84, 1:116, 2:8, 2:140-141, 3:79- 80, 3:87, 3:89, 3:91, 3:93, 3:169-170, 5:137
Instant Messaging (IM)	4:88
Institute of Electrical and Electronics Engineers (IEEE)	2:15
Intel	1:84, 1:97, 3:73-74, 3:170
Internet Protocol (IP)	2:10, 2:17-18, 3:57, 3:66, 3:163, 5:93, 5:105, 5:107, 5:111-112, 5:142
Interprocess Communication	3:25
iOS Version	1:11, 1:45, 1:51, 1:62, 1:64, 1:68, 2:11, 2:14, 2:34, 3:48, 3:89
iOS Versions	1:11, 2:34
IP address	2:18, 3:66, 3:163, 5:105, 5:107, 5:111-112, 5:142
IP addresses	5:107, 5:142
iPad Mini	2:13

iPhone 5	1:46, 4:178
iPhone 5s	1:46
IPS	1:31, 1:44, 1:46, 5:151
IPv6	5:105, 5:107, 5:111-112
istat	2:78, 2:199, 4:55
iTunes	1:50, 1:54, 1:61, 1:64, 1:66, 1:68, 1:74, 3:87, 3:170, 4:158, 4:172, 5:64

J

Jabber	3:16
Jailbreak	1:44, 1:50-52, 2:32, 3:18, 4:28, 5:86
JAMF Casper Suite	1:109
Java	3:170, 5:83-84, 5:87, 5:92-93
JavaScript Object Notation (JSON)	1:16, 1:18, 2:53, 4:150, 4:175
John the Ripper	5:150-151, 5:158
JPEG	2:169, 2:171, 4:94

K

KeepAlive key	3:21
KeePass	1:89
KeRanger	5:89
keylogger	3:35, 3:75, 5:87, 5:114
kill	2:68, 5:84
Knockknock	3:17
KnownNetworks	2:21
Kumar-In-The-Mac (KitM)	5:88

L

LastSession.plist	4:55, 4:57
Layer	1:83, 2:107, 3:85, 3:148, 4:99, 5:87
Leaf Nodes	2:161, 2:182, 2:184, 2:186, 2:189, 2:193
Legal Authority	5:67
libewf	1:39
Library folder	1:24
Linux	1:14, 1:32, 1:58, 3:75, 5:128, 5:135
lists	1:15-18, 1:111, 2:19, 2:50, 2:60, 2:81, 2:149, 3:30, 3:32, 3:41, 3:43, 3:58, 3:141, 4:8, 4:14, 4:59, 4:102, 4:114, 4:132, 4:135,

5:72-73, 5:105, 5:107, 5:117, 5:128, 5:134,
 5:136, 5:151, 5:194
 Little Endian 1:101, 1:103, 1:105, 1:107, 2:45, 3:74
 localtime 2:9, 3:87, 3:149, 4:36, 4:47, 4:59, 4:79,
 4:97, 4:112, 4:114, 4:126, 4:133, 4:135,
 4:137, 4:143, 4:151, 5:24, 5:26
 Lock 1:31-32, 1:36, 1:40, 1:47, 1:49-50, 1:59-60,
 1:70, 1:83, 1:85, 1:92, 1:111, 1:117, 2:11,
 2:14, 2:141-142, 2:147, 2:152, 2:154-160,
 2:170, 2:174-175, 2:186, 2:191, 2:197,
 2:201, 2:204, 3:16, 3:29, 3:58, 3:70, 3:118,
 3:139, 3:166, 4:18, 4:62, 4:93, 4:139,
 4:179, 5:8, 5:12-13, 5:24, 5:26-27, 5:151-
 152, 5:165, 5:186-187, 5:197
 Lockdown File 1:59-60, 1:70, 2:11
 Lockdown key 1:70
 Locking 1:47-48, 1:59-60, 1:70, 2:13-14, 5:13
 Locks 1:32, 1:40, 1:50, 1:83, 1:92, 1:117, 2:147,
 2:152, 2:155-160, 2:170, 2:174-175, 2:186,
 2:197, 3:58, 5:165, 5:186-187
 Logging 2:100, 3:46, 3:110-118, 3:143, 3:150, 5:117
 Logical Acquisition 1:35, 1:57, 5:24
 Logical Extraction 1:21
 Logical Volume (LV) 1:40, 1:111
 Logical Volume Family (LVF) 1:111
 Logical Volume Group (LVG) 1:111
 logs 2:11, 2:23, 2:72, 2:86, 2:89, 2:100, 3:23-
 24, 3:39, 3:87, 3:89, 3:91, 3:98-101,
 3:103-104, 3:107, 3:110, 3:115-119, 3:121-
 122, 3:133, 3:135, 3:137, 3:139, 3:141,
 3:144, 3:146, 3:150, 3:153-154, 3:162-164,
 3:166-168, 3:175-177, 3:179-180, 5:87,
 5:112, 5:129, 5:151, 5:186
 Loopback 5:112
 lsof 5:107, 5:114
 LSSharedFileList 3:43
 LTE 1:105, 2:68, 2:89, 2:100, 2:174, 3:57, 3:75,
 3:114-115, 3:127, 3:144, 4:33, 4:40, 5:24-
 25, 5:114, 5:131, 5:147
 Lurker 2:32, 5:86

M

Mac Absolute Time	1:12, 5:97
MAC address	2:15, 2:18, 3:69, 3:158, 3:176, 4:180, 5:26-28, 5:109, 5:111-112, 5:179
Mac Memory Reader	5:128
Mac Memoryze	5:128
mac_pslist	5:138, 5:140
mac_pstree	5:140
Mach Object (Mach-O)	3:74
Mach-O	3:74, 3:115, 5:128
MacQuisition	1:21, 1:33-34, 5:130-131
magic	2:155, 2:157, 3:125, 4:118
Maiyadi	1:52
Malware	3:17-18, 3:23-24, 3:75, 3:170, 5:82-93, 5:99, 5:124, 5:136
Malware, Detection	5:136
man	1:40, 1:102, 1:117, 3:21, 3:102, 3:126, 5:104, 5:107, 5:118
Manifest.db	1:72-73
manifest.mbdb	1:72
Manifest.mbdb	1:72
Manifest.plist	1:70, 1:76, 3:14
Map Nodes	2:182-183
Map Records	2:182
Mapping	4:19, 4:169, 5:104
mapsdata	4:170-172
mask	2:17, 2:97
Master Boot Record (MBR)	1:83, 1:90, 1:97-100, 1:102-103, 5:155
matches	2:127
matching	3:56, 3:163, 4:40, 4:119, 4:133, 4:171
mbox	4:72, 4:77-78
MCC	5:24
mdfind	2:115-116, 2:120-121
mdimport	2:108
mdls	2:116-117, 2:119-120
Media Access Control (MAC)	2:15, 2:18, 3:69, 3:158, 3:176, 3:179, 4:180, 5:26-28, 5:109, 5:111-112, 5:179
Memory acquisition	1:33, 5:126-131, 5:133
Memoryze	5:128
Metadata	1:21, 1:72-73, 1:83, 1:85, 2:67, 2:70-71, 2:74, 2:81-82, 2:86, 2:107-108, 2:110, 2:115-117, 2:119-121, 2:140, 2:147, 2:158-

	159, 2:161, 2:166-169, 2:194, 2:201, 2:203-204, 3:11, 3:55, 3:64, 3:84, 4:15, 4:49-50, 4:54, 4:59, 4:63, 4:71, 4:74, 4:78-79, 4:82, 4:90-91, 4:120-121, 4:123, 4:142, 4:161-163, 5:38, 5:40-41, 5:93, 5:177, 5:181, 5:188, 5:210
Methods	1:31, 1:47, 1:54, 1:76, 3:18, 5:13, 5:86
Microsoft	1:12, 1:27, 2:45-46, 2:62, 3:23-24, 3:49-50, 3:170, 4:13, 4:22, 4:70, 5:33, 5:57
Microsoft Active Directory (AD)	3:158-159
MIME	4:94
mmls	1:102
MNC	5:24
Mobile Device	1:10, 2:30, 2:143
Mobile Time Machine File System (MTMFS)	5:201, 5:203
MobileSync Folder	1:61, 1:66
modules	3:75, 5:21-23
Morcut	5:87
MSAB	5:64
Multimedia Messaging Service (MMS)	2:100, 3:121, 5:173

N

nanopasses.sqlite3	4:181
Near Field Communication (NFC)	4:178
NetBIOS	2:17
Netcat	1:33
Netstat	5:105, 5:109, 5:142
Network Attached Storage (NAS)	5:111
Network Interface	2:15, 2:17, 5:112, 5:142
Network Interface Card (NIC)	3:158
nmap	5:128
Node	1:73, 2:24, 2:28, 2:78, 2:82, 2:114, 2:147, 2:154, 2:161-171, 2:181-187, 2:189-194, 2:196, 3:49-50, 3:64, 3:124-125, 3:167, 4:112, 4:114, 5:41, 5:104, 5:107, 5:147-148
Normalization	1:79, 3:98
Notes	1:48, 1:60-61, 1:97, 1:105, 1:107, 2:140, 2:154-160, 2:162, 2:165-169, 3:11, 3:18, 3:35, 3:75, 4:70, 4:77, 4:129-133, 4:135, 4:137, 4:139-143, 4:145, 5:79

notes.sqlite 4:131-133
 Notification 1:52, 3:29, 3:32, 3:169, 4:96
 NowSecure 4:28
 NSKeyedArchiver 1:73, 2:44, 2:50-51, 2:53, 2:55, 2:57-58,
 3:24, 3:32, 3:43, 4:99
 NSPreservationReason 5:40
 NT File System (NTFS) 1:83, 5:171

O

Objects 1:15, 2:50, 2:55, 2:58, 2:152, 2:154, 2:159,
 2:161
 Open Firmware 1:31-32, 1:84, 3:73
 OpenGL 4:18
 OpenSSH 1:53, 3:18, 3:60, 3:163
 OSXAuditor 3:17
 osxautoruns 3:17
 OSXPmem 5:129
 OSXPMem 5:129
 Outlook 4:70
 overrides.plist 3:60, 3:64
 owner 1:36, 1:38-40, 2:21, 2:117, 2:168, 2:194,
 2:197, 4:126
 Oxygen 1:74, 5:64
 Oxygen Forensics Suite 1:74, 5:64

P

pac4mac 3:17
 packages 1:51, 3:170, 4:158, 5:135
 Packet Filter 3:57
 Pages 1:102, 2:107, 2:132, 5:33, 5:67, 5:79,
 5:104, 5:185
 pangu 1:52
 parameters 1:38-39, 3:122, 5:204
 partition 1:21, 1:31-32, 1:44-45, 1:48, 1:50, 1:52-55,
 1:82-85, 1:87-90, 1:92, 1:94, 1:97-100,
 1:102-103, 1:105, 1:107, 1:109-110, 1:115-
 117, 1:119, 2:141, 3:18, 5:155, 5:163, 5:167
 partition table 1:84, 1:97-100, 1:105, 1:107
 Passcode 1:47-48, 1:59-60, 1:70, 2:13-14, 5:13

Passes	2:43, 4:147-151, 4:153, 4:181
Passware	5:150, 5:152
passwd	1:30
password	1:31-32, 1:38, 1:40, 1:44, 1:47, 1:57, 1:60, 1:74, 1:76-77, 2:23, 2:26, 2:29, 3:10-11, 3:14-16, 3:35, 3:63, 3:124-125, 3:139, 4:8, 4:28, 4:137, 4:139, 5:66, 5:134, 5:145-148, 5:150-153, 5:155, 5:158-160, 5:175
Password Cracking	5:145-147, 5:150-151, 5:159-160
PBKDF2	5:147-148
PDF	1:32, 1:48, 1:60, 1:97, 1:105, 1:107, 2:30, 2:32, 2:107, 2:152-171, 3:17, 3:56, 5:86
Pegasus	5:90
Performance Optimization With Enhanced RISC - Performance Computing (PowerPC)	1:84, 3:73-74
Perl	2:29, 3:63, 3:121
Permissions	1:24, 2:40, 2:67, 2:168, 2:194, 2:197, 3:64, 4:20, 4:22, 4:24-25, 4:32, 4:37, 5:34, 5:93, 5:129
persistence	3:18, 3:23-24
pfctl	3:57
Phishing	5:83
Photo Databases	4:161
PhotoData	4:161-163, 4:165
PhotoStreamsData	4:165
Physical Acquisition	1:55, 1:58, 2:12-13, 3:87, 4:18, 4:20, 4:49, 5:17
Physical Volume (PV)	1:111
PIN	1:49, 1:59
ping	3:58
Plaintext	2:81, 3:7, 3:9, 3:16, 3:100, 3:102, 4:8, 4:67, 4:132, 5:45, 5:159
PList Editor	1:16
Plists	1:15-16, 2:19, 2:50, 2:60, 3:32, 4:14, 4:59
plutil	1:16, 1:18-20, 2:53, 2:81, 5:40
PMem	5:129
pointer	2:182
Pointer Records	2:182
Policy	2:26, 3:122, 4:24
POST	3:53
Power Nap	5:164
praudit	3:98, 3:124, 3:127

Prefetch	2:119
Preflight	5:186-187
printers.conf	3:53
Privacy	2:30, 3:57, 4:21-24, 5:18, 5:61, 5:85
Privilege	1:33, 1:44, 1:50, 2:23-24, 2:86, 3:8, 3:111, 3:124, 3:162, 3:165, 3:168, 5:129, 5:205
Properties	2:58, 2:60, 2:86, 2:117, 3:141, 3:148, 4:180
Property List Files (Plist)	1:16-18, 1:20, 1:27-28, 1:32, 1:53, 1:59, 1:62, 1:67-68, 1:70, 1:76, 2:7, 2:9-15, 2:17, 2:19-21, 2:24, 2:26-30, 2:34, 2:38, 2:43- 46, 2:48, 2:50-51, 2:53, 2:55, 2:57, 2:60, 2:62, 2:70, 2:74, 2:81, 2:110, 3:10, 3:13-14, 3:21-24, 3:26, 3:30, 3:32, 3:37, 3:39, 3:41, 3:43-44, 3:48-50, 3:53, 3:58, 3:60-61, 3:64, 3:67, 3:69-72, 3:75, 3:77, 3:81, 3:83- 84, 3:93, 3:133, 3:150, 3:167, 3:176, 4:7-8, 4:19, 4:25-28, 4:45-46, 4:55, 4:57, 4:59, 4:61, 4:65, 4:72, 4:82, 4:99, 4:103-105, 4:121, 4:161, 4:163, 4:170-171, 4:174, 4:180, 5:40, 5:71-75, 5:95, 5:98-99, 5:164, 5:167, 5:169, 5:185, 5:188, 5:194, 5:196- 197, 5:199
Proprietary	1:10, 1:27, 1:31, 1:61, 3:55, 3:102, 4:9, 4:67, 4:141, 4:170-171
Protective MBR	1:83, 1:102-103
proxy	4:28, 4:178, 4:180
ps	1:32, 5:118
plist	5:138, 5:140
pstree	5:140
public key	3:11, 3:66
pwd	3:149
Python	2:62, 2:88, 2:91, 2:100, 2:122, 3:14, 5:7- 16, 5:69, 5:93, 5:135-138

Q

qtkitserver	3:26
Quarantine	2:66, 2:68-71, 2:74, 2:165, 2:201, 2:203, 4:82, 4:159, 5:92, 5:95-98

R

random	4:8, 5:13, 5:128
Random Access Memory (RAM)	1:32, 5:118, 5:127-128, 5:130-131, 5:133, 5:159
Ransomware	5:89
RecentlyClosedTabs.plist	4:57
Recovery	1:35, 1:76, 1:85, 1:87, 1:100, 1:109, 1:113, 2:160, 3:121, 3:170, 5:152, 5:179
Reduced Instruction Set Computing (RISC)	3:73
Registry	1:16, 3:23-24, 4:180-181
Regulator	3:179
Rekall	5:129, 5:135-136
Reminders	1:48, 4:21-23, 4:101-102, 4:112, 4:114, 4:179
Remote Access	3:57-58
Repair	2:191
Reporting	3:170
Response	1:3, 1:33, 3:8, 3:75, 4:11, 4:50, 5:103-105, 5:107, 5:109, 5:111-112, 5:114, 5:116-118, 5:120-122, 5:124, 5:130, 5:133
Restricted	1:60, 2:23
Reverse Engineering (RE)	2:68
Ruby	2:29
Run As	5:112

S

Safari	1:15, 1:61, 2:23, 2:66, 2:69-70, 2:120, 2:135, 2:144, 2:166, 3:50, 3:74, 4:19, 4:21, 4:41, 4:43-47, 4:49-50, 4:52, 4:54-55, 4:57, 4:59, 4:61-65, 4:67, 4:84, 4:94, 4:132, 5:97
Safety Table	1:102-103
Salt	5:147-148
Sandbox	1:27-28, 1:44, 3:46, 3:103, 4:13, 4:81, 5:71, 5:96
Sandboxing	1:27-28, 1:44, 3:103, 4:13
Santoku	1:58
SCDMA	5:24
SD Card	1:90, 3:143

Searches 2:107, 2:109, 2:116, 2:119-121, 2:192, 4:45,
 4:170, 4:172-173, 5:75, 5:77
Sectors 1:40, 1:92, 1:98-99, 1:103
Secure Enclave 1:46
Secure Shell (SSH) 1:52, 1:54, 1:58, 3:18, 3:60, 3:66, 3:163,
 4:28
Sednit 5:91
segment 5:131
Sensitivity 2:174, 5:188
Service Set IDentifier (SSID) 2:18-19, 2:21, 3:117
SFTP 1:58
sh 4:28
shadow 5:147-148
ShadowHashData 5:147-148
Shell 3:8, 3:60, 4:34, 5:129
Short Message Service (SMS) 4:90-91, 4:93-94
SHUTDOWN_TIME 3:154
Signature 1:14, 1:16, 1:103, 1:105, 1:116, 2:128, 2:174,
 3:74, 3:102, 4:140, 5:98-99, 5:155
Simple Mail Transfer Protocol (SMTP) 2:34
SkipPaths 5:167
Sleep 3:153, 3:156, 5:134, 5:164
Sleuth Kit (TSK) 1:102, 2:78, 2:82, 2:140, 2:142, 2:175,
 2:178, 2:181-184, 2:187, 2:189, 2:191-194,
 2:199-200, 2:202
SleuthKit 1:102, 2:78, 2:82, 2:140, 2:142, 2:175,
 2:181-184, 2:187, 2:189, 2:191-194, 2:200,
 2:202
slice 1:36, 1:40, 1:45, 1:85
SMB 3:64, 3:144
SMS.db 4:90-91, 4:93-94
SMTP 2:34
snapshot 1:67, 2:148-151, 2:159-160, 3:172, 4:18-19,
 4:63, 5:67, 5:138, 5:164, 5:167, 5:173,
 5:177, 5:181, 5:183, 5:185-188, 5:190,
 5:194, 5:197, 5:199, 5:201-205, 5:207,
 5:209-210
Snapshot Contents 5:185
Social Engineering (SE) 3:179
socket 5:105, 5:111
Sofacy 5:91
Software Updates 3:170, 5:164
Solaris 3:119

Solid-state Drive (SSD)	1:31-32, 1:115
Sparse Bundle	1:109, 1:116, 5:155, 5:167, 5:192, 5:194, 5:199
Sparse Disk Image	1:109, 1:116
Spotlight Store	2:110, 2:112
Spyware	5:87, 5:90
SQL	1:14, 2:38, 4:33, 4:126
SQLite	1:14-15, 1:72, 2:36, 2:38, 2:40, 2:88, 2:91, 2:119, 3:10, 3:14, 3:33-34, 3:87, 4:9, 4:20, 4:22, 4:28, 4:30-31, 4:33-41, 4:45-47, 4:49-50, 4:59, 4:63, 4:76, 4:79, 4:90-91, 4:93, 4:96, 4:99, 4:112, 4:114, 4:120, 4:126, 4:131-133, 4:135, 4:137, 4:139-142, 4:145, 4:151, 4:153, 4:161-163, 4:181, 5:9- 10, 5:13, 5:21-23, 5:41, 5:43, 5:47-48, 5:96
SSH	1:52-54, 1:58, 3:18, 3:21, 3:43, 3:50, 3:60, 3:66, 3:163, 4:28
SSID	2:18-19, 2:21, 3:117, 3:176-177
Standards	5:128
Status.plist	1:67
StealthBit	5:85
Store-V1	2:110
Store-V2	2:110, 2:112
Store.sqlite	4:131-132, 4:135, 4:137, 4:139-142, 4:145, 5:13
stream	1:61, 2:165-166, 2:169, 3:17, 4:165
Strings	1:15, 1:21, 1:74, 2:47, 2:86, 3:16, 3:35, 3:55, 4:170, 4:173, 5:134, 5:159
Structured Query Language (SQL)	1:14, 2:38, 4:33, 4:126
su	3:124, 3:165, 5:129
subkeys	5:167
sudo	1:32, 1:36-40, 1:99, 2:29, 2:114, 3:50, 3:165, 5:129
sudoers	3:165
Sumuri Recon	5:133
SUSE	1:111, 5:167, 5:188
SuspendState.plist	4:55, 4:61
swap	1:87, 5:128, 5:134
Switch	1:97, 4:18
Synalyze It!	2:174, 5:52
Syslog	3:102-104, 3:106-108
SYSLOG	3:102-104, 3:106-108
system_profiler	2:7, 5:120, 5:122

T

tag	1:97, 3:24, 3:73, 3:122, 4:151, 5:95, 5:118, 5:127
Target	1:32
Target Disk Mode (TDM)	1:31-32
TCC.db	4:20-24, 4:32
Tech Note 1150	2:140, 2:142, 2:175, 2:181-184, 2:187, 2:189, 2:191-194, 2:200, 2:202
terminal log	3:163
Termination	3:125
TextEdit	2:93, 5:33, 5:45, 5:67, 5:79
TFTP	5:186
Thunderbolt	1:31, 2:17, 3:37
Ticket	4:148-149, 4:153
Time Capsule	1:25, 5:163-164, 5:167, 5:192
Time Machine	2:86, 2:110, 2:148, 3:11, 3:135, 3:173, 5:162-167, 5:169, 5:171, 5:173, 5:175, 5:177, 5:179, 5:181, 5:183, 5:185, 5:188, 5:190, 5:192, 5:194, 5:196-197, 5:199, 5:201-205, 5:207, 5:209-210, 5:212, 5:214
Time To Live (TTL)	3:102
Timestamp	1:12, 1:15, 1:21, 1:59, 1:62, 1:66-67, 1:73, 2:8, 2:26, 2:32, 2:36, 2:43, 2:46, 2:72, 2:74, 2:76, 2:100, 2:102, 2:108-110, 2:117, 2:119, 2:147, 2:157, 2:160, 2:178, 2:194, 2:196, 3:9, 3:39, 3:53, 3:69, 3:81, 3:83-84, 3:87, 3:98, 3:106, 3:114, 3:125, 3:143, 3:148, 3:150, 3:154, 3:166, 3:176, 4:36, 4:39, 4:41, 4:47, 4:50, 4:64-65, 4:79, 4:91, 4:96, 4:99, 4:114, 4:121, 4:133, 4:135, 4:137, 4:142, 4:151, 4:159, 4:162, 4:175, 4:180, 5:8, 5:24, 5:26, 5:41, 5:43, 5:48, 5:67, 5:72-73, 5:97, 5:116-117, 5:167, 5:173, 5:183, 5:188, 5:199
tmutil	2:148-149, 5:166, 5:205, 5:207, 5:209-210, 5:212
TN1150	2:140, 2:142, 2:175, 2:181-184, 2:186-187, 2:189, 2:191-194, 2:200-202
tokens	3:125-126
Touch ID	1:46-49, 2:14, 4:149, 5:13
Transmission Control Protocol (TCP)	5:105, 5:107
Trojan	5:84-87

TrueCrypt	3:170
trusted	1:59, 3:119
TTL	1:101, 1:103, 1:105, 1:107, 2:45, 3:74, 3:102, 4:71
Twitter	1:1, 1:120, 2:1, 2:34, 2:38, 2:40, 3:1, 3:182, 4:1, 4:22-23, 4:183, 5:1, 5:98, 5:161
Two-Factor	3:70, 5:69-70
Type 0x0001	2:193
Type 0x0002	2:193
Type 0x0003	2:193
Type 0x0004	2:193

U

UARFCN	5:24
UberCab	4:11, 4:22
UEFI	1:105, 1:107
UFED Link Analysis	5:64
underscore	2:128
Unicode	1:107, 2:189, 2:192, 2:196, 2:204, 4:93, 4:170
uniquesize	5:205
Unix epoch	1:12, 2:26, 2:71-72, 3:53, 3:108, 3:154, 4:36, 5:167, 5:183
Uptime	5:116
User Account File	2:24, 2:26, 5:107
User Context	1:32, 3:25
User Datagram Protocol (UDP)	5:105, 5:107
UTC	1:12, 2:45, 2:119, 2:167, 2:196, 3:98, 3:107-108, 4:36
utf-16	2:132-133

V

Vendors	3:141
Verification	5:196
Video	2:144, 3:110-118, 4:16, 4:21-22, 4:94, 4:132, 4:165-166
Virtual Memory	5:134
Virtual Volumes (VVOLs)	1:111, 1:113
Virus	5:87, 5:95-99, 5:101

VMEM	5:138
VMware	4:158, 5:138
VNC	3:61, 3:63, 3:163
Volatility	5:135-138, 5:140, 5:142
volume header	2:141, 2:174, 2:178
vulnerability	5:83-84

W

Wallet	2:43, 4:147-151, 4:153, 4:181
WatchOS	1:10, 3:28, 4:179
Webcam	5:87
WebKit	1:12, 4:54, 5:97
WEP	5:112
whatis	3:133
WhatsApp	2:147, 4:16-17, 4:21
wheel	3:149, 5:129
whoami	5:116
WifiLocation	5:26
Wireless	2:12, 2:19, 2:21, 3:158, 3:176-177, 3:179, 4:25, 5:112
WireLurker	2:32, 5:86
Wireshark	3:170
WPA	5:112
WPA2	5:112
wtmp	3:102-103, 5:117

X

XACT	2:10, 2:21, 4:12, 4:19, 4:34, 4:63, 4:172, 5:25
xattr	2:68, 2:71, 2:81, 2:128, 2:165-166, 5:40, 5:179, 5:183, 5:188
Xcode	1:16-17, 1:19-20, 1:24, 2:50, 3:170, 3:176, 4:24, 4:28, 4:121, 5:97, 5:212
XOR	2:29, 3:63
XPC	3:25-26, 3:60
XProtect	5:95, 5:98-99
XRY	5:64

Z

ZACCOUNT	2:36, 4:133, 4:135
ZAUTHORIZATION	2:40
zip	1:68, 2:86, 2:91, 2:95, 2:97, 2:100, 3:100-101, 4:132, 4:137, 4:140, 5:19, 5:129
Zydia	1:52

"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