USING APPLE “BUG REPORTING” FOR FORENSIC PURPOSES

Heather Mahalik
Guest Starring: Mattia Epifani
Primary Scripting: Adrian Leong

OSDFCON
OCTOBER 2019
APPLE PROFILE AND LOGS

• Apple provides “a web-based tool that developers can use to report issues with Apple software and services, request enhancement to APIs and tools and track the status of their feedback”

• To correctly use this tool and submit Apple relevant information to identify the issue, it is mandatory to “Collect and attach any relevant logs”
• The Apple web page “Profiles and Logs” contains instructions about how to extract logs from different Apple operating systems, including Mac OS X, iOS, tvOS and WatchOS

• Some logs (e.g. Crash Logs) are automatically generated by the operating system during its execution while others (e.g. sysdiagnose) can be generated with specific user actions

• Moreover, some logs require the installation of a profile on the device (e.g. Disk Space Diagnostics and Battery Life)
Using Apple “Bug Reporting” for forensic purposes

• We wrote a document describing our research into these logs
• This document is freely available from https://www.for585.com/sysdiagnose
• We also developed various scripts to parse some of the files available during sysdiagnose acquisition
• These scripts are available from GitHub https://github.com/cheeky4n6monkey/iOS_sysdiagnose_forensic_scripts
Automatically generated by the operating system when an application crashes

Can be used to understand the **conditions under which the application terminated**

/private/var/mobile/Library/Logs/CrashReporter/

/private/var/root/Library/Logs/CrashReporter/
COLLECTING THE LOGS

Methods...
1 - Using an iOS device manager tool
2 – Sync the iOS device with iTunes

<table>
<thead>
<tr>
<th>OS</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>macOS</td>
<td>/Users/&lt;username&gt;/Library/Logs/CrashReporter/MobileDevice/[Device_Name]/</td>
</tr>
<tr>
<td>Windows</td>
<td>C:\Users&lt;username&gt;\AppData\Roaming\Apple Computer\Logs\CrashReporter\MobileDevice[Device_Name]\</td>
</tr>
</tbody>
</table>
3 - Using idevicecrashreport tool

```
C:\Windows\System32\cmd.exe
D:\mattia\Downloads\libimobiledevice (1)\libimobiledevice\win7-x64\bin\idevicecrashreport -e -k CrashLogs
Copy: /stacks+routined-2019-05-02-221628.ips
```
4 - Using Elcomsoft iOS Forensic Toolkit
5 - Last effort... Using AIRDROP
What’s Coming Next - The Choice is Yours...
Unlike Crash Logs, sysdiagnose logs are not executed and written automatically by the operating system.

The generation must be triggered manually by the user.

There are two documented procedures to generate sysdiagnose logs:

1. By simultaneously pressing and releasing both volume buttons + the Side (or Top) button for 1 to 1.5 seconds
2. By using AssistiveTouch

The sysdiagnose logs can be extracted from an iOS device using the same methods described for the extraction of Crash Logs.
GENERATING SYSDIAGNOSE – IN THE BACKGROUND…
Open source

Developed with Python3 standard libraries (e.g. plistlib)

Avoids third party libraries as forensic workstations may not be connected to the Internet

Written/prototyped on Ubuntu 16.04 LTS running Python 3.5

14 scripts (so far) with 3 categories of script:

- iOS Configuration
- Network Info
- App Info
# SYSDIAGNOSE PARSING SCRIPTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sysdiagnose-sys.py</code></td>
<td>Extracts OS info from logs/SystemVersion/SystemVersion.plist</td>
</tr>
<tr>
<td><code>sysdiagnose-networkprefs.py</code></td>
<td>Extracts hostnames from logs/Networking/preferences.plist</td>
</tr>
<tr>
<td><code>sysdiagnose-networkinterfaces.py</code></td>
<td>Extracts network config info from logs/Networking/NetworkInterfaces.plist</td>
</tr>
<tr>
<td><code>sysdiagnose-mobilecontainermanager.py</code></td>
<td>Extracts uninstall info from logs/MobileContainerManager/containermanagerd.log.0</td>
</tr>
<tr>
<td><code>sysdiagnose-mobilebackup.py</code></td>
<td>Extracts backup info from logs/MobileBackup/com.apple.MobileBackup.plist</td>
</tr>
<tr>
<td><code>sysdiagnose-mobileactivation.py</code></td>
<td>Mobile Activation Startup and Upgrade info from logs/MobileActivation/mobileactivationd.log.*</td>
</tr>
<tr>
<td><code>sysdiagnose-wifi-plist.py</code></td>
<td>Extracts Wi-Fi network values from WiFi/com.apple.wifi.plist Use <code>-t</code> option for TSV output file</td>
</tr>
<tr>
<td><code>sysdiagnose-wifi-icloud.py</code></td>
<td>Extracts Wi-Fi network values from WiFi/iCloud.apple.wifid.plist Use <code>-t</code> option for TSV output file</td>
</tr>
<tr>
<td><code>sysdiagnose-wifi-net.py</code></td>
<td>Extracts Wi-Fi network names to categorized TSV files from WiFi/wifi <em>.</em> log</td>
</tr>
<tr>
<td><code>sysdiagnose-wifi-kml.py</code></td>
<td>Extracts Wi-Fi geolocation values and creates a KML from wifi*.log</td>
</tr>
<tr>
<td><code>sysdiagnose-uuid2path.py</code></td>
<td>Extracts GUID and path info from logs/tailspindb/UUIDToBinaryLocations</td>
</tr>
<tr>
<td><code>sysdiagnose-net-ext-cache.py</code></td>
<td>Extracts app name &amp; GUID info from logs/Networking/com.apple.networkextension_cache.plist Use <code>-v</code> option to print GUID info</td>
</tr>
<tr>
<td><code>sysdiagnose-appconduit.py</code></td>
<td>Extracts connection info from logs/AppConduit/AppConduit.log.*</td>
</tr>
<tr>
<td><code>Sysdiagnose-appupdates.py</code></td>
<td>Extracts update info from logs/appinstallation/AppUpdates.sqlite.db*</td>
</tr>
</tbody>
</table>
# WIFI PLIST (I)

<table>
<thead>
<tr>
<th>SSID</th>
<th>BSSID</th>
<th>NETUSAGE</th>
<th>COUNTRYCODE</th>
<th>LASTJOINED</th>
<th>LASTAUTOJOINED</th>
</tr>
</thead>
</table>
WIFI PLIST (II)

BSSID/MAC: 80:2a:a8:1a:03:e9

SSID / Network Name (exact match): foobar

Average Location - Coordinates
Lot: 47.25649 to: 47.25695
Lon: -87.256243 to: -87.256244
Search Radius Tolerance(+/ degrees): 0.010

Network Location

Click for interactive map

---

Map | Net ID | SSID | Name | Type | First Seen | Most Recently | Crypto | Est. Lat |
--- | --- | --- | --- | --- | --- | --- | --- | --- |
map | 80:2a:a8:1a:03:E9 | ITTIG-CNR | Infra | 2016-09-23T00:00:00.000Z | 2018-01-25T09:00:00.000Z | | 43.79183678 | 11.22836018 | 6
Step 1 - WIFI KML Script
Step 2 - WIFI KML Script

MacBook-Air-di-Mattia:sysdiagnose mattiaeipin1$ python3 sysdiagnose-wifi-kml.py -i ../Test_Data/WiFiManager/wifi.log
Running sysdiagnose-wifi-kml.py v2019-06-08 Initial Version

====================================================================

Found 89 valid didUpdateLocation instances in ../Test_Data/WiFiManager/wifi.log
Found 10 valid _WiFiManagerGeoTagNetwork instances in ../Test_Data/WiFiManager/wifi.log
Found 7 valid _WiFiManagerLocationManagerCallback instances in ../Test_Data/WiFiManager/wifi.log
Found 2 valid _WiFiLocaleManagerLocationManagerCallback instances in ../Test_Data/WiFiManager/wifi.log
Found 68 valid WiFiLocaleManagerCheckLocale instances in ../Test_Data/WiFiManager/wifi.log
Found 7 valid _WiFiDeviceManagerAttemptNetworkTransition instances in ../Test_Data/WiFiManager/wifi.log
Found 2 valid _WiFiDeviceManagerScanPreviousNetworkChannel instances in ../Test_Data/WiFiManager/wifi.log
Found 0 valid WiFiManagerCopyCurrentLocation instances in ../Test_Data/WiFiManager/wifi.log

Logged 177 locations to wifi-buf-locations.kml output file
Ignored 0 malformed log entries
MOBILE INSTALLATION LOGS


MacBook-Air-di-Mattia:MobileInstallation mattiaepifani$ python3 mib_parser.sql.py

iOS Mobile Installation Logs Parser
By: @AlexisBrignoni
Web: abrignoni.com

Logs processed: 2
Lines processed: 2535

Total apps: 29
Total installed apps: 29
Total uninstalled apps: 0
Total historical app reports: 29
Total system state events: 6
What’s Coming Next – Be Smart About Your Choice...
INSTALLING PROFILES ON THE DEVICE

• Other logs can be generated by installing specific “profiles” on the device
• Profiles can be downloaded from the Apple website
• The most interesting profiles from a digital forensics perspective are:
  – Battery Life
  – Disk Space Diagnostics (FS Metadata)
  – WiFi (may already be there)
Just Tell Me The Proper Order Already

If the “iTunes encryption” is haunting you

• The “Reset Network Settings” will scrub the com.apple.wifi.plist
• The “Reset All Settings” may scrub other logs – need more testing here
• Get at least Sysdiagnose first

When “PowerLogs” matter

• i.e. What happened in the last 10 mins on the device?
• Here, you would install the Battery Life profile first
• Regular acquisition methods impact the logs
• APOLLO can be used to parse them

Yes, you are installing a profile on the device

• We do this all of the time with logical extractions

Documentation and reasoning are key!
INSTALLING A PROFILE ON A DEVICE (I)
INSTALLING A PROFILE ON A DEVICE (II)
Considerations

• What if the device is locked?
• Is this forensically sound?
• What will your organizations/departments think?
• How can we get this peer reviewed?

Profile Updates/Changes
  – File System profile went MIA 😞

A full file system extraction gets some logs already
  – Cellebrite Premium, CAS and GrayKey
  – Sysdiagnose is NOT one of the logs captured by these methods – do it after

Sysdiagnose is essentially us conducting “live forensics” on a Apple device
  – Research, Test, and Validate
REFERENCES

- Using Apple “Bug Reporting” for forensic purposes
  https://www.fors585.com/sysdiagnose
- Apple Bug Reporting
- Apple Profiles and Logs
- Understanding Crashes and Crash Logs
- Understanding and Analyzing Application Crash Reports
- Demystifying iOS Application Crash Logs
  https://www.raywenderlich.com/2805-demystifying-ios-application-crash-logs
- The ultimate diagnostic tool: sysdiagnose
  https://eclecticlight.co/2016/02/06/the-ultimate-diagnostic-tool-sysdiagnose/
- More useful information gleaned from sysdiagnose
  https://eclecticlight.co/2016/02/08/more-useful-information-gleaned-from-sysdiagnose/
- Running tools within sysdiagnose individually
  https://eclecticlight.co/2016/02/08/running-tools-within-sysdiagnose-individually/
- iOS Mobile Installation Logs
  https://dfir.pubpub.org/pub/e5xlbw88
**SYSDIAGNOSE TOOLS**

- **Libimobiledevice**  [https://www.libimobiledevice.org/](https://www.libimobiledevice.org/)
- **DB Browser for SQLite**  [https://sqlitebrowser.org/](https://sqlitebrowser.org/)
- **Elcomsoft iOS Toolkit**  [https://www.elcomsoft.com/eift.html](https://www.elcomsoft.com/eift.html)
- **iOS Sysdiagnose Forensic Scripts**  [https://github.com/cheeky4n6monkey/iOS_sysdiagnose_forensic_scripts](https://github.com/cheeky4n6monkey/iOS_sysdiagnose_forensic_scripts)
- **iOS Mobile Installation Logs Parser**  [https://github.com/abrignoni/iOS-Mobile-Installation-Logs-Parser](https://github.com/abrignoni/iOS-Mobile-Installation-Logs-Parser)
- **APOLLO**  [https://github.com/mac4n6/APOLLO](https://github.com/mac4n6/APOLLO)
Questions?

Mattia Epifani
- Digital Forensics Analyst
- CEO @ REALITY NET – System Solutions
- SANS Instructor, FOR585 / FOR500

Heather Mahalik
- Senior Director of Digital Intelligence
- Cellebrite
- SANS Senior Instructor, FOR585 / FOR500

mattia.epifani@realitynet.it
@mattiaep
http://www.linkedin.com/in/mattiaepifani
http://www.realitynet.it
http://blog.digital-forensics.it

Heather@cellebrite.com
@heathermahalik
https://it.linkedin.com/in/heathermahalik
www.smarterforensics.com/blog
www.cellebrite.com/en/blog