Poisoned Apples:
Current State of iOS Malware detection

Matthias Frielingsdorf | Trail of Bits (iVerify)
Who am I

Matthias Frielingsdorf

Former: Mobile Security Team @ Deutsche Bahn

Now: iOS Security Researcher at Trail of Bits (iVerify)

Twitter: @helthydriver
iOS Exploits?

Started with iOS 14.4

Older versions often patched

More Information is not available

3rd party reports are sometimes available

iOS 16.4.1 and iPadOS 16.4.1

Released April 7, 2023

**iOSurfaceAccelerator**

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: An app may be able to execute arbitrary code with kernel privileges. Apple is aware of a report that this issue may have been **actively exploited**.

Description: An out-of-bounds write issue was addressed with improved input validation.

CVE-2023-28206: Clément Lecigne of Google's Threat Analysis Group and Donncha Ó Cearbhaill of Amnesty International's Security Lab

**WebKit**

Available for: iPhone 8 and later, iPad Pro (all models), iPad Air 3rd generation and later, iPad 5th generation and later, and iPad mini 5th generation and later

Impact: Processing maliciously crafted web content may lead to arbitrary code execution. Apple is aware of a report that this issue may have been **actively exploited**.

Description: A use after free issue was addressed with improved memory management.

WebKit Bugzilla: 254797

CVE-2023-28205: Clément Lecigne of Google’s Threat Analysis Group and Donncha Ó Cearbhaill of Amnesty International’s Security Lab

https://support.apple.com/en-us/HT213720
iOS Threat Notifications 2021

iMessage + Email to associated AppleID

Visible on https://appleid.apple.com

Help available at: https://securityplanner.consumerreports.org/tool/emergency-resources

https://support.apple.com/en-us/HT212960
iOS CVEs - „actively exploited“

CVE-2016-4657  CVE-2016-4655  CVE-2016-4656  CVE-2018-4442
CVE-2020-27932  CVE-2019-6225  CVE-2020-27930  CVE-2020-27950
CVE-2021-31010  CVE-2021-30983  CVE-2021-30663  CVE-2021-30883
CVE-2021-30858  CVE-2021-30860  CVE-2021-30661  CVE-2021-1870  CVE-2022-42827
CVE-2021-30807  CVE-2021-30665  CVE-2021-1879  CVE-2021-1782  CVE-2022-42856
CVE-2021-30666  CVE-2021-1871  CVE-2021-30761  CVE-2021-30762  CVE-2023-23529
Today

1. Study of Recent Examples of 0/1 - Click Malware
2. Detection Capabilities - App
3. Detection Capabilities - Companion / MDM
4. Detection Capabilities - Forensic
5. Synthesis
6. Improving & Further Research
1. Study of recent examples of Malware
2016 Pegasus v1

Infection Vector
WebKit

Targets
Humans Rights Activists
Journalists

Detection & Technical Analysis
CitizenLab and Lookout

CVEs
CVE - 2016 - 4657  WebKit
CVE - 2016 - 4655  Infoleak
CVE - 2016 - 4656  Kernel

Detection
Access to one Infection URL

IOCs
URLs
Files
Processes

Attribution
NSO

https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/
### 2019 - A campaign against Uygures

<table>
<thead>
<tr>
<th>iOS 9</th>
<th>iOS 10</th>
<th>iOS 11</th>
<th>iOS 12</th>
<th>iOS 13</th>
<th>iOS 14</th>
<th>iOS 15</th>
<th>iOS 16</th>
</tr>
</thead>
</table>

#### Infection Vector
WebKit

#### Targets
Uygures in Nepal

#### Detection & Technical Analysis
Google TAG & Google Project 0

#### CVEs
- CVE - 2017 - 2505 WebKit
- CVE - 2017 - 7064 WebKit
- CVE - 2018 - 4122 WebKit
- CVE - 2018 - Unk. WebKit
- CVE - 2018 - 4438 WebKit
- CVE - 2018 - 4442 WebKit
- CVE - 2019 - 6217 WebKit
- CVE - 2019 - 7287 Kernel
- CVE - 2017 - 13861 Kernel
- CVE - 2019 - 6225 Kernel

#### Detection
Detection of Infiltration Network Implant + Exploit Download

#### IOCs
- File URLs Process

#### Attribution
Unknown (Maybe China)

---

https://googleprojectzero.blogspot.com/2019/08/implant-teardown.html
https://googleprojectzero.blogspot.com/2019/08/jsc-exploits.html
https://googleprojectzero.blogspot.com/2019/08/a-very-deep-dive-into-ios-exploit.html
2021 Pegasus v2

Infection Vector
- iMessage

Targets
- Humans Rights Activists
- Journalists, Politicians

Detection & Technical Analysis
- CitizenLab, Amnesty International & Google Project 0

CVEs
- CVE - 2021 - 30860
- iOS 10: iMessage
- iOS 11: VoWiFi
- iOS 12: iMessage
- iOS 13: PhotoStream
- iOS 13: IMTransco.

Detection
- Forensic Analysis of infected iPhones
- Infiltration / CnC Infrastructure

IOCs
- Files
- URLs
- Processes
- Email Addresses
- iCloud Addresses

Attribution
- NSO

https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html
2021 - Predator

iOS 9  iOS 10  iOS 11  iOS 12  iOS 13  iOS 14  iOS 15  iOS 16

Infection Vector
WebKit

Targets
Meta Manager, Politician, Journalist

Detection & Technical Analysis
CitizenLab

CVEs
iOS 13  Unknown
iOS 14  Unknown

Detection
Forensic Analysis

IOCs
Files
Processes
URLs
Shortcut

Attribution
Cytrox

2022 - Hermit

Infection Vector
- Sideloaded App

Targets
- Kazakhstan, Italy

Detection & Technical Analysis
- Google Project 0, Google Tag, Lookout (Android)

CVEs
- CVE - 2018 - 4344: Lightspeed
- CVE - 2019 - 8605: SockPort2
- CVE - 2020 - 3837: TimeWaste
- CVE - 2020 - 9907: AveCesare
- CVE - 2021 - 30883: Clicked2
- CVE - 2021 - 30983: Clicked3

Detection
- Access to the App and infection URL

IOCs
- App URLs
- Provisioning Profile

Attribution
- RCS Labs

https://googleprojectzero.blogspot.com/2022/06/curious-case-carrier-app.html
https://www.lookout.com/blog/hermit-spyware-discovery
2023 - Reign

iOS 9  iOS 10  iOS 11  iOS 12  iOS 13  iOS 14  iOS 15  iOS 16

**Infection Vector**
- Calendar Events

**Targets**
- Civil Society

**Detection & Technical Analysis**
- CitizenLab & Microsoft Threat Intelligence

**CVEs**
- iOS 14  ENDOF DAYS

**Detection**
- Access to Loader & Forensic

**IOC s**
- Files
- Processes
- URLs

**Attribution**
- QuaDream

https://citizenlab.ca/2023/04/spyware-vendor-quadream-exploits-victims-customers/
2023 - Jamf Threat Labs Report

Infection Vector
Unknown

Targets
Journalists

Detection & Technical Analysis
Jamf Threat Labs

CVEs
- iOS 14: Unknown
- iOS 15: Unknown

Detection
Forensic Analysis

IOCs
- Files
- Processes

Attribution
NSO, Unknown

https://www.jamf.com/blog/threat-advisory-mobile-spyware-continues-to-evolve/
# 2023 - Pegasus v3

<table>
<thead>
<tr>
<th>iOS 9</th>
<th>iOS 10</th>
<th>iOS 11</th>
<th>iOS 12</th>
<th>iOS 13</th>
<th>iOS 14</th>
<th>iOS 15</th>
<th>iOS 16</th>
</tr>
</thead>
</table>

## Infection Vector
- Homekit, iMessage, FindMy

## CVEs
- iOS 15: FINDMYPWN
- iOS 15: LATENTIMAGE
- iOS 16: PWNYOURHOME

## Targets
- Mexico Civil Society

## Detection
- Forensic Analysis

## Detection & Technical Analysis
- CitizenLab

## IOCs
- Files
- Processes
- Crashlog

## Attribution
- NSO

---

iOS CVEs - map "exploited" to reports

<table>
<thead>
<tr>
<th>CVE-2016-4657</th>
<th>CVE-2016-4655</th>
<th>CVE-2016-4656</th>
<th>CVE-2018-4442</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2020-27932</td>
<td>CVE-2019-6225</td>
<td>CVE-2020-27930</td>
<td>CVE-2020-27950</td>
</tr>
<tr>
<td>CVE-2021-31010</td>
<td>CVE-2021-30983</td>
<td>CVE-2021-30663</td>
<td>CVE-2021-30883</td>
</tr>
<tr>
<td>CVE-2021-30858</td>
<td>CVE-2021-30860</td>
<td>CVE-2021-30661</td>
<td>CVE-2021-1870</td>
</tr>
<tr>
<td>CVE-2021-30807</td>
<td>CVE-2021-30665</td>
<td>CVE-2021-1879</td>
<td>CVE-2021-1782</td>
</tr>
<tr>
<td>CVE-2021-30666</td>
<td>CVE-2021-1871</td>
<td>CVE-2021-30761</td>
<td>CVE-2021-30762</td>
</tr>
</tbody>
</table>
# Target Data

<table>
<thead>
<tr>
<th>Sample</th>
<th>App List</th>
<th>Crash Logs</th>
<th>Files</th>
<th>Network</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hermit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Jamf Report</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Reign</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Pegasus</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Predator</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Target Malware Categories

- Malicious
- Apps
- Profiles
- Known Implants
- Unknown Implants
- App*
- MDM
- Companion
- Forensic

* Under the assumption of a sandboxed app
2. Detection Capabilities - App
Jailbreaks

App Jailbreaks
- Need an Application to start the jailbreak
- Semi (Untethered)
- unc0ver, Fugu15, Cheyote (not released)

Boot Jailbreaks
- Jailbreak during device boot
- Tethered
- Checkra1n, Palera1n

The process of jailbreaking

1. Exploit
2. Elevation of Privileges
3. Shell Environment
4. App Store + Tweaks (Cydia)
## Malware vs Jailbreaks

<table>
<thead>
<tr>
<th>Malware</th>
<th>Jailbreaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker oriented</td>
<td>User - oriented</td>
</tr>
<tr>
<td>0-Days</td>
<td>Known Exploits</td>
</tr>
<tr>
<td>Stealth</td>
<td>Open</td>
</tr>
<tr>
<td>Extract Data + Control</td>
<td>Liberate Device + Apps</td>
</tr>
<tr>
<td>(Un)detectable</td>
<td>Detectable</td>
</tr>
<tr>
<td>Only few known examples</td>
<td>Many known examples</td>
</tr>
<tr>
<td>Targeted</td>
<td>Mass Market</td>
</tr>
</tbody>
</table>
Jailbreak Detection in 2023

Detection Mechanisms

- Files / Folders on disk
- Changes on the device (remounted root partition)
- Process / SharedCache Injection
- Disabled Security Mechanisms
- Protocol Handlers

Reporting …

- „Jailbreak“ detected
- One event triggers detection
Malware detection with Apps

We can use everything that Apple allows us to do ;)

- Check for the existence of a file
  -> We can't read it (so we don't know if it's really an issue)
  -> "com.apple.CrashReporter.plist" a bad example (also in Beta)

- Install a VPN Profile / Proxy to inspect network traffic
  -> We can't decrypt traffic
  -> If the infrastructure is known we can use it to detect infections

- Similar to jailbreaks we can detect the absence of security mechanisms or manipulation of the app
2. Detection Capabilities - App - Data

- Method
- App List
- Crash Logs
- Files
- Network
- Processes

App*

✓ **
 ✓

* Under the assumption of a sandboxed app  ** Only for known file path
2. Detection Capabilities - App - Malware

- Malicious
  - App*
  - MDM
  - Companion
  - Forensic

- Apps
- Profiles
  - Known Implants
  - Unknown Implants

✓**

* Under the assumption of a sandboxed app  ** Only for known file path
3. Detection Capabilities - Companion / MDM
Mobile Device Management - MDM

Allows companies to control certain behavior of iOS devices

Features depend on:

- Apples MDM Protokoll
  
https://github.com/apple/device-management

- Supervised vs. Non-Supervised Devices
  
https://support.apple.com/en-gb/guide/deployment/dep0f7dd3d8/web
Mobile Device Management - MDM

Detection:
- DeviceInformation (iOS Version…)
- ConfigurationProfileList
- CertificateList
- ProvisioningProfileList
- InstalledApplicationList
Mobile Device Management - MDM

Prevention:
- Trust new proprietary in-house apps developers
- Users accept untrusted TLS certificates
- Allow a configuration profile to be installed (Supervised Only)
Companion App - lockdownd

Uses Apple’s Lockdown Daemon (lockdownd) on a desktop

- Used by Xcode, Apple Configurator, iTunes, …
- Starts services that are accessible via USB connections

Can be accessed via commercial tools like iMazing or FOSS like libimobiledevice

- [https://libimobiledevice.org](https://libimobiledevice.org)

-> Currently developed by [@nikias](https://twitter.com/nikias)

-> Available for Linux, Mac & Windows
# libimobiledevice - Supported (iOS 13.5)

<table>
<thead>
<tr>
<th>Lockdown Service</th>
<th>Status</th>
<th>iOS Version</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.apple.afc</td>
<td>Done</td>
<td>13.5.1</td>
<td>Fully implemented except a few details.</td>
</tr>
<tr>
<td>com.apple.companion_proxy</td>
<td>Partly</td>
<td>11</td>
<td>Full implemented except a few details.</td>
</tr>
<tr>
<td>com.apple.iosdiagnostics.relay</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.diagnostics_relay</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented</td>
</tr>
<tr>
<td>com.apple.mobile.heartbeat</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.house_arrest</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.insecure_notification_proxy</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.installation_proxy</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.mobile_image_mounter</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobile.notification_proxy</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
</tbody>
</table>
libimobiledevice - Supported (iOS 13.5)

<table>
<thead>
<tr>
<th>Lockdown Service</th>
<th>Status</th>
<th>iOS Version</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.apple.mobileactivationd</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobilebackup</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobilebackup2</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.mobilesync</td>
<td>Partly</td>
<td>11</td>
<td>Partly implemented.</td>
</tr>
<tr>
<td>com.apple.preboardservice</td>
<td>Partly</td>
<td>11</td>
<td>Partly implemented.</td>
</tr>
<tr>
<td>com.apple.preboardservice_v2</td>
<td>Partly</td>
<td>11</td>
<td>Partly implemented.</td>
</tr>
<tr>
<td>com.apple.springboardservices</td>
<td>Partly</td>
<td>11</td>
<td>Partly implemented.</td>
</tr>
<tr>
<td>com.apple.syslog_relay</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
<tr>
<td>com.apple.webinspector</td>
<td>Done</td>
<td>11</td>
<td>Fully implemented.</td>
</tr>
</tbody>
</table>
libimobildevice - unsupported (iOS 13.5)

<table>
<thead>
<tr>
<th>Lockdown Service</th>
<th>Status</th>
<th>iOS Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.apple.atc</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.atc2</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.bluetooth.BTPacketLogger</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.crashreportcopymobile</td>
<td>Done</td>
<td>11</td>
<td>*Is implemented - but unsure if fully supported</td>
</tr>
<tr>
<td>com.apple.crashreportmover</td>
<td>Done</td>
<td>11</td>
<td>*Is implemented - but unsure if fully supported</td>
</tr>
<tr>
<td>com.apple.idamd</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.misagent</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.mobile.assertion_agent</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.mobile.file_relay</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.mobile.MCInstall</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.os_trace_relay</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.pcapd</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.PurpleReverseProxy.Conn</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.PurpleReverseProxy.Ctrl</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>com.apple.streaming_zip_conduit</td>
<td>None</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>
### libimobildevice - Tools

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
<th>Detection Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>idevice_id</td>
<td>List attached devices or print device name of given device</td>
<td></td>
</tr>
<tr>
<td>idevicebackup2</td>
<td>Create or restore backup for devices (idevicebackup for iOS &lt; 4)</td>
<td></td>
</tr>
<tr>
<td>idevicecrashreport</td>
<td>Retrieve crash reports from a device</td>
<td>Retrieve Crashlogs &amp; Sysdiagnose Information</td>
</tr>
<tr>
<td>idevicedate</td>
<td>Display the current date or set it on a device</td>
<td></td>
</tr>
<tr>
<td>idevicedebug</td>
<td>Interact with the debugserver service of a device</td>
<td></td>
</tr>
<tr>
<td>idevicedebugserverproxy</td>
<td>Proxy a debugserver connection from a device for remote debugging</td>
<td></td>
</tr>
<tr>
<td>idevicediagnostics</td>
<td>Interact with the diagnostics interface of a device</td>
<td></td>
</tr>
<tr>
<td>ideviceenterrecovery</td>
<td>Make a device enter recovery mode</td>
<td></td>
</tr>
<tr>
<td>ideviceimagemounter</td>
<td>Mount disk images on the device</td>
<td></td>
</tr>
<tr>
<td>ideviceinfo</td>
<td>Show information about a connected device</td>
<td></td>
</tr>
<tr>
<td>*ideviceinstaller</td>
<td>*Manage Apps on the device</td>
<td>*List Apps</td>
</tr>
<tr>
<td>idevicename</td>
<td>Display or set the device name</td>
<td></td>
</tr>
<tr>
<td>ideviciplatformproxy</td>
<td>Post or observe notifications on a device</td>
<td></td>
</tr>
<tr>
<td>idevicepair</td>
<td>Manage host pairings with devices and usbmuxd</td>
<td></td>
</tr>
<tr>
<td>ideviceprovision</td>
<td>Manage provisioning profiles on a device</td>
<td>Retrieve Provisioning Profiles -&gt; 3rd Party Apps</td>
</tr>
<tr>
<td>idevicescreenshot</td>
<td>Gets a screenshot from the connected device</td>
<td></td>
</tr>
<tr>
<td>idevicesetlocation</td>
<td>Simulate location on device</td>
<td></td>
</tr>
<tr>
<td>idevicesyslog</td>
<td>Relay syslog of a connected device</td>
<td>Live Syslog Information</td>
</tr>
</tbody>
</table>
3. Detection Capabilities - Companion / MDM - Data

<table>
<thead>
<tr>
<th>Method</th>
<th>App List</th>
<th>Crash Logs</th>
<th>Files</th>
<th>Network</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>App*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MDM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Companion</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Backup</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sysdiagnose</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Under the assumption of a sandboxed app  ** Only for known file path
3. Detection Capabilities - Companion / MDM - Malware

<table>
<thead>
<tr>
<th>Malicious</th>
<th>Apps</th>
<th>Profiles</th>
<th>Known Implants</th>
<th>Unknown Implants</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

* Under the assumption of a sandboxed app  ** Only for known file path
4. Detection Capabilities - Forensics
4. Detection Capabilities - Forensics

**File System:**
- iTunes Backup (via lockdownd)
- Full FileSystem extraction (Jailbreak required)

**Diagnostic Information:**
- CrashLogs (via lockdownd)
- Sysdiagnose (via lockdownd)
Forensic Data - iTunes Backup

Can be created on multiple ways

- Commercial Forensic Tool (Cellebrite, Elcomsoft, Magnet…)
- iMazing (there is a free version available)
- iTunes / Finder
- libimobildedev
device

Encrypted & Unencrypted Backups
- Encrypted Backups contain a lot more sensitive information
Forensic Data - Analyzing Backups

Commercial Tools (Cellebrite, Magnet, …)
- Will help to decode data and display contents
- Not focused on malware detection

Multiple opensource tools available like:
- https://github.com/avibrazil/iOSbackup (Python)
- Allow easy access to data. Parsing iOS files (plist, sqlite databases, NSKeyedArchiver… might be cumbersome

Trainings available to navigate iOS Backups / FS data
e.g. SANS FOR 518 (Created by Sarah Edwards)
Forensic Data - Analyzing Backups - MVT

Developed by Amnesty International Tech Lab

Created to make iOS forensic artifact analysis a lot easier
Focus on Spyware Analysis
- https://mvt.re/
- https://github.com/mvt-project/mvt

Works on iTunes Backups & FileSystem dumps, supports STIX2 for IOCs

* Amnesty Investigations IOCs - https://github.com/AmnestyTech/investigations
Forensic Data - Analyzing Backups - MVT

On backup analysis records are extracted
- Some sample records* are:

<table>
<thead>
<tr>
<th>Record</th>
<th>Specific Files</th>
<th>Detection Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>applications.json</td>
<td>Info.plist, iTunesMetadata.plist</td>
<td>List of Apps, Non AppStore Apps</td>
</tr>
<tr>
<td>configuration_profiles.json</td>
<td>Configuration Profiles</td>
<td>Configuration Profiles</td>
</tr>
<tr>
<td>shortcuts.json</td>
<td>/private/var/mobile/Library/Shortcuts/Shortcuts.sqlite</td>
<td>Might be used for persistence</td>
</tr>
<tr>
<td>interaction_c.json</td>
<td>/private/var/mobile/Library/CoreDuet/People/interactionC.db</td>
<td>Interaction with installed Apps</td>
</tr>
<tr>
<td>manifest.json</td>
<td>Manifest.db</td>
<td>Some FilePaths</td>
</tr>
<tr>
<td>os_analytics_ad_daily.json</td>
<td>/private/var/mobile/Library/Preferences/com.apple.osanalytics.addaily.plist</td>
<td>Data Usage by Processes</td>
</tr>
<tr>
<td>datausage.json</td>
<td>/private/var/wireless/Library/Databases/DataUsage.sqlite</td>
<td>Network Data Usage by Processes, Bundle Identifier</td>
</tr>
<tr>
<td>profile_events.json</td>
<td>Configuration Profiles</td>
<td>Changes on Configuration Profiles</td>
</tr>
<tr>
<td>shutdown_log.json</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcc.json</td>
<td>/private/var/mobile/Library/TCC/TCC.db,</td>
<td>Access to Microphone, Camera, Location</td>
</tr>
</tbody>
</table>

- Additionally records for Domains/URLs & FileSystem dumps

* [https://docs.mvt.re/en/latest/ios/records/](https://docs.mvt.re/en/latest/ios/records/)
Forensic Data - Crashes & Sysdiagnose

iOS keeps logs app and kernel crashes; can be seen in the settings app:

- **Settings** -> **Data Privacy & Security**
- **Analysis & Improvements** -> **Analysis Data**

Sysdiagnose has to be triggered manually & will be available in the same place

- iPhone X key combination: Volume Up + Down + Power for 0.7 Seconds

Forensic Data - Crashes & Sysdiagnose

1) Key combination 2) Wait ~ 5 min 3) Check folder

sysdiagnose file is roughly 100 - 500 MB of data

You can sent the file via share dialog or
- sync with iTunes
 (~/Library/Logs/CrashReporter/MobileDevice/)
- copy crashlogs with libimobiledevice
Forensic Data - Sysdiagnose

Let's have a look:
Forensic Data - Sysdiagnose

Contains basically everything interesting you want to look at ;)
- Process Names
- Mount / Partition Information
- App Names, Updates & Uninstalls
- Information on Backups

Excellent Paper available at: http://www.for585.com/sysdiagnose

Tools to parse sysdiagnose data:
https://github.com/cheeky4n6monkey/iOS_sysdiagnose_forensic_scripts
https://github.com/abrignoni/iLEAPP
4. Detection Capabilities - Forensics - Data

<table>
<thead>
<tr>
<th>Method</th>
<th>App List</th>
<th>Crash Logs</th>
<th>Files</th>
<th>Network</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>App*</td>
<td></td>
<td></td>
<td>✓**</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MDM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Companion</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backup</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sysdiagnose</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* Under the assumption of a sandboxed app  ** Only for known file path
4. Detection Capabilities - Forensics - Malware

<table>
<thead>
<tr>
<th>Malicious</th>
<th>Apps</th>
<th>Profiles</th>
<th>Known Implants</th>
<th>Unknown Implants</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>App</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓**</td>
<td>✓</td>
</tr>
<tr>
<td>MDM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Companion</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forensic</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Under the assumption of a sandboxed app  ** Only for known file path
5. Synthesis - Bringing it all together
# Manual vs. (Semi) - Automatic Detections

<table>
<thead>
<tr>
<th></th>
<th>Automatic</th>
<th>Semi - Automatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apps</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Profiles</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Known Implants</td>
<td>✓**</td>
<td>✓</td>
</tr>
<tr>
<td>Unknown Implants</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

* Under the assumption of a sandboxed app  ** Only for known file path
Manual vs. (Semi) - Automatic Detections

All the tools are available to detect our known malware samples

- some: can be detected automatically
- all: can be detected semi-automatically with a companion app / forensic analysis

There are manual tools available to detect unknown malware samples if
- a device is infected
- a proper forensic & sysdiagnose analysis is executed
Status Quo - Implementation

Most companies will only do the automatic detections (MDM + App) as this is available and known.

To improve we need:
- companion apps and tools to support analysis
- skilled people that can do an analysis
- people to train other people in malware detection
- to make data extraction and analysis easier
What to do if your phone be.a.es w.i.d.y

Contact an expert:

Apple recommends in their threat notifications:  
https://securityplanner.consumerreports.org/tool/emergency-resources

Amnesty International & CitizenLab are known to be experts in the field.  
Feel free to contact us at Trail of Bits / iVerify  
https://www.iverify.io/contact /
6. Improving the detection
Improving Jailbreak Detection - 2023

Reporting ...

„Jailbreak“ detected
One event triggers detection

Better Reporting

Don’t detect Jailbreaks, detect mal. behavior!
Every event is reported for analysis
Differentiate active & inactive Jailbreaks
App Jailbreak vs. Boot Jailbreak
Validate findings (iOS Version, Device Type...)

Better Reporting: Every event is reported for analysis, differentiate active & inactive Jailbreaks, app vs. boot jailbreak, validate findings with iOS version and device type.
Improving Malware Detection

<table>
<thead>
<tr>
<th>Apple</th>
<th>Companies</th>
<th>IoT Experts + Defensive Companies</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Code Quality and Exploitation)</td>
<td>Crash log &amp; Forensic Analysis</td>
<td>Crash log &amp; Forensic Analysis</td>
</tr>
<tr>
<td>Endpoint security capabilities</td>
<td>Companion App</td>
<td>Training on Malware detection</td>
</tr>
<tr>
<td>FileSystem and Process Access</td>
<td>Monitoring network traffic</td>
<td>Set Focus on Malware Detection</td>
</tr>
</tbody>
</table>
Further Research

Combining EMM / MTD with Crash Log & Forensic Analysis

Combining iOS and macOS Agents

iOS Backups / FileSystem / Syslog Data
Conclusion

Apple’s walled garden raises the bar for exploitation every year

We need more focus on malware detection

Improvements have to be make on several levels

It is not possible to detect _new_ malware with an app on the device

We need more companion apps for forensic analysis

We need more training & skilled people
Conclusion

“…The capability to target and monitor the private activities of entire populations in real time.“

Ian Beer (2019)
Additional Information + Contact Data

Contact me on

Spyware Information (Twitter)

Twitter

LinkedIn
Thank you!