The Journal of Open Engineering

Against the Law: Countering Lawful Abuses of Digital Surveillance

bunnie Huang, Edward Snowden

Published on: Jul 21, 2016
Updated on: Jul 14, 2019
DOI: 10.21428/12268

As a class, investigative journalists face a heightened risk of retaliation for their work. Conservative figures from the Committee to Protect Journalists show at least 1,240 journalists killed for their work since 1992; this figure includes only deaths where the motive could be confirmed[1]. Unfortunately, recent advances in technology are giving rise to a circumstance where journalists are being betrayed by their own tools: their smartphones can be transformed into tracking devices. Governments and powerful political institutions are actively exploiting the unwitting emissions of phones, leaving journalists, activists, and rights workers in a new position of constant vulnerability. This work aims to give journalists a tool for observing when their smart phones are creating emissions, even when the devices are supposed to be in airplane mode. We propose to accomplish this via direct introspection of signals controlling the phone's radio hardware. The Introspection Engine will be an open source, user-inspectable and field-verifiable module attached to an existing smart phone that makes no assumptions about the trustworthiness of the phone's operating system.

Introduction and Problem Statement

Today, journalists, activists, and rights workers occupy a position of vulnerability. A great portion of this vulnerability originates from the opacity of modern devices: there are simply no tools available through which one can determine what is happening beneath the glass and icons, preventing the development of a natural understanding of dangerous device states. We cannot secure what we cannot inspect.

According to the Committee to Protect Journalists' figures, reporters covering politics are more likely to be killed for their work than any others, even those covering war[1]. Given the power and resources at the disposal of the politicians being investigated, these journalists can face unusually sophisticated threats. A new "digital arms" industry has risen to develop hardware and software transforming the omnipresent smartphones of citizens into ideal tracking devices, and this industry has few qualms about selling such technology to those seeking to violate human rights [2]. This is a grave development facing those who speak truth to power, as the relationship between people and their phones is far more intimate than with traditional computers. Carried in pockets and purses, smartphones and their increasingly-robust sensor arrays bear witness day and night to a modern user's life. By contrast, modern laptops typically lack even a basic GPS module and struggle to maintain powered for more than half a day. As serious as the hacking of a business or personal laptop can

be, such hardware limitations create natural restrictions on the extent of a compromise.

Even without hacking, mobile phones invisibly transmit dangerously rich records about their owners' private activities. Recounting the full scope of this threat to confidentiality – colloquially described as the "metadata problem" – is beyond the scope of our paper, but those with a specific interest may find it addressed at book length in Bruce Schneier's recent Data and Goliath [3]. It is sufficient for our purposes to acknowledge merely carrying a phone connected to the cellular network delivers a comprehensive record of one's movements and call activity to tower operators and service providers, and these records may be retained for decades [4].

Compounding these technical risks, the United States has promoted a notoriously lax regulatory regime in which such records enjoy little meaningful legal protection by narrowing citizens' "reasonable expectation of privacy" [5]. This doctrine has spread to other countries through a kind of legal contagion, establishing a harmful global norm in which alarmingly comprehensive records of the private activities of individuals are routinely available to agents of government, even in circumstances where this is undesirable and socially dangerous. While there is little doubt such capabilities are abused in places like Syria, China, Russia, and the like, this is not a problem exclusive to authoritarian states. Indeed, some of the most prominent abuses related to the targeting of journalists' smartphones and related records in recent years have occurred in Australia [6] and Canada [7]. Admitting to precisely the problematic access of a journalist's records with which this work is concerned, Australian Federal Police Commissioner Andrew Colvin sought to downplay the severity of the breach, stating, "we use metadata on nearly all of our investigations. It is a very common tool we use." [8]

The value of such clandestine access to targets' records and smartphones is perhaps most presciently illustrated by the US National Security Agency in a Top Secret document, as summarized in Figure 1.

Figure 1: Top Secret slides extracted from the Snowden Archive illustrating one intelligence agency's perspective on metadata and location services offered by a major US brand [9]

Reporter Marie Colvin's 2012 death is a tragic reminder of how real this vulnerability can be. A lawsuit against the Syrian government filed in 2016 alleges she was deliberately targeted and killed by Syrian government artillery fire. The lawsuit describes how her location was discovered in part through the use of intercept devices that monitored satellite-dish and cellphone communications.[10]

As groups like the University of Toronto's Citizen Lab uncover sophisticated hacking campaigns explicitly targeting the phones of journalists [11], we have a moral obligation to consider what can be done. Currently, they are asked to trust their safety to nothing more reliable than the presence or absence of icons and other settings purporting to represent whether the device is in "airplane mode." But airplane mode is no defense; for example, on iPhones since iOS 8.2, GPS is active in airplane mode. Furthermore, airplane mode is a "soft switch"–the graphics on the screen have no essential correlation with the hardware state. Malware packages, such as were observed being planted on journalists' phones by Citizen Lab, can be designed to activate radios without any indication from the user interface; trusting a phone that has been hacked to go into airplane mode is like trusting a drunk person to judge if they are sober enough to drive.

This work aims to give journalists the tools to understand and, eventually, control when their smart phones are tracking or disclosing their location via radio frequency emissions. We propose to accomplish this via direct introspection of signals controlling the phone's radio hardware. The Introspection Engine will be an open source, user-inspectable and field-verifiable module attached to an existing smart phone that makes no assumptions about the trustworthiness of the phone's operating system.

Approach and Goals

Numerous researchers and extensive corporate resources have been dedicated to the task of building a more secure smart phone. However, smartphones are extremely complex and present a large, porous attack surface. Furthermore, even a perfectly secure phone will not save a reporter from "victim-operated" exploits such as spearphishing. Eliminating this vector is complicated by the fact that effective

reporters must communicate with a diverse array of sources who may intentionally or unintentionally convey a malware payload to the reporter.

As a result, this work starts with the assumption that a phone can and will be compromised. In such a situation, a reporter cannot take the UI status at face value. Instead, we aim to provide field-ready tools that enable a reporter to observe and investigate the status of the phone's radios directly and independently of the phone's native hardware. We call this direct introspection, a term we derive from techniques pioneered by the space[12] and supercomputing[13] industries to improve fault-tolerance and to detect erroneous operation of hardware.

Our work proposes to monitor radio activity using a measurement tool contained in a phone-mounted battery case. We call this tool an Introspection Engine. The Introspection Engine has the capability to alert a reporter of a dangerous situation in real-time. The core principle is simple: if the reporter expects radios to be off, alert the user when they are turned on.

Our introspection engine is designed with the following goals in mind:

1. Completely open source and user-inspectable ("You don't have to trust us")
2. Introspection operations are performed by an execution domain completely separated from the phone's CPU ("don't rely on those with impaired judgment to fairly judge their state")
3. Proper operation of introspection system can be field-verified (guard against "evil maid" attacks and hardware failures)
4. Difficult to trigger a false positive (users ignore or disable security alerts when there are too many positives)
5. Difficult to induce a false negative, even with signed firmware updates ("don't trust the system vendor" – state-level adversaries with full cooperation of system vendors should not be able to craft signed firmware updates that spoof or bypass the introspection engine)
6. As much as possible, the introspection system should be passive and difficult to detect by the phone's operating system (prevent black-listing/targeting of users based on introspection engine signatures)
7. Simple, intuitive user interface requiring no specialized knowledge to interpret or operate (avoid user error leading to false negatives; "journalists shouldn't have to be cryptographers to be safe")
8. Final solution should be usable on a daily basis, with minimal impact on workflow (avoid forcing field reporters into the choice between their personal security and

being an effective journalist)

This work is not just an academic exercise; ultimately we must provide a field-ready introspection solution to protect reporters at work. Although the general principles underlying this work can be applied to any phone, reducing these principles to practice requires a significant amount of reverse engineering, as there are no broadly supported open source phone solutions on the market. Thus we focus on a single phone model, the 4.7” iPhone 6 by Apple Inc., as the subject for field deployment. The choice of model is driven primarily by what we understand to be the current preferences and tastes of reporters. It has little to do with the relative security of any platform, as we assume any platform, be it iOS or Android, can and will be compromised by state-level adversaries.

Faraday Cages Alone are Not an Option

A Faraday cage is an electromagnetic shield constructed from a metal mesh or foil. When constructed and used properly, it is an extremely effective surveillance countermeasure. The United States government has long relied upon them as part of the TEMPEST standard, which regulates protection levels for classified information processing spaces. Although conceptually simple, their efficacy hinges upon proper construction and routine maintenance. Even small holes and gaps can lead to exploitable emissions. These holes and gaps develop at seams where two surfaces connect, or where conduit (bearing power or communications lines) must penetrate the cage due to the natural degradation of cage materials from environmental factors. They must be detected and patched (typically with copper tape) to prevent unintentional emissions. Even the “gold-plated” room- and trailer-scale implementations in real-world field use by the government at covert sites must be subjected to complex annual certification and accreditation procedures. Since detecting anything smaller than a catastrophic leak can be quite difficult, these procedures require a few hours of specialist work with sensitive signal generation and detection equipment.

While there is a robust commercial market in bag-style Faraday cages for phones, we could identify no public data confirming their reliability over time under real-world field-use conditions. Yet even if the fragility of Faraday cages – evidenced so clearly by the painstaking maintenance required of climate-controlled government facilities – was somehow not a factor despite the abuse journalistic equipment is subjected to in

field environments, there's a more fundamental problem: a phone confined to a magic bag is as useful as an expensive brick.

While Faraday cages are simple in concept, any slot or hole in the cage, intentional or unintentional, will leak radiation. For example, creating an aperture for photography and control of the camera function would irreparably compromise the efficacy of the Faraday cage. Thus, any effective Faraday cage would run counter to the basic requirement that the phone be usable as a journalistic tool. The purpose of direct introspection is to enable journalists to carry a single, compact tool that can take photographs, shoot video, record audio, and serve as a word processor without betraying their position in the field. Forcing a reporter to choose between their safety – that is, keeping their phone in a Faraday cage – and taking photographs in the field violates goal number 8. Furthermore, asking reporters deep in war zones to carry a separate camera, audio recorder, and word processor to avoid surveillance is also not a practical option.

Finally, part-time use of a Faraday cage without any additional countermeasures can be problematic for reporters. Faraday cages only isolate the phone from electromagnetic radiation, so malware can still log unaffected sensors such as the microphone. More significantly, even brief removal of the phone from the Faraday cage can provide sufficient connectivity to determine position and even transmit partial logs. Thus, part-time use of a Faraday cage can create a false sense of security, which can lead to risky behavior that can eventually compromise the reporter's position or their contacts.

Instead, a reporter may be well-advised to use a Faraday cage in conjunction with the novel methods presented here that detect the presence of threats. A Faraday cage would mitigate any failure of direct introspection, while our introspection methods can alert journalists to the presence of attempted malware transmissions, and in some cases actively mitigate or prevent damage.

Methods & Results

The first step toward executing this work was to visit the Hua Qiang electronics markets of Shenzhen to collect samples and documentation for evaluation. These markets are used for the trade and practice of iPhone repair; as such, it is a rich source of spare parts and repair manuals. The repair manuals frequently contain

detailed blueprints of the iPhone 6, which were used to assist the reverse engineering effort.

Based on the phone model selection and available documentation, we can enumerate the radio interfaces available:

- Cellular modem – 2G/3G/4G
- Wifi / BT
- GPS
- NFC (Apple Pay)

Although our work can be extended to input systems such as the IMU (inertial measurement unit), barometer, microphone and camera, to focus the effort we restrict our exploration to only RF interfaces that can directly betray a user's location. Note that a camera can be defeated by obscuring the lens; as such the final physical design of our Introspection Engine will likely include a feature to selectively obscure the rear camera lens.

Methods that Do Not Meet our Criteria

Numerous semi-intrusive countermeasures were considered along the way to our current solution, including but not limited to RF spectrum monitoring, active jamming, and the selective physical isolation or termination of antennae. Semi-intrusive countermeasures would require minimal modification to the phone itself, which is desirable as it simplifies field deployment and could even enable reporters to perform the modifications without any special tools. Unfortunately, all of these methods were deemed to be inadequate, as discussed in the following paragraphs.

RF spectrum monitoring consists of building an external radio receiver that can detect transmissions emanating from the phone's radios. In some cases, it was hypothesized that the receiver could be as trivial as an RF power monitor within the anticipated radio bands. A simple example of such monitoring already exists in the form of novelty lights that flash based on parasitic power extracted from the GSM antennae. The problems with this approach are that 1) only active transmissions from the radio can be reliably detected, and 2) malware that passively records the user's position and delivers it as a deferred payload when the radios are intentionally activated cannot be detected. Furthermore, this approach is subject to spoofing; false positives can be triggered by the presence of nearby base stations. Such false alarms can confuse the

user and eventually lead the user to be conditioned to ignore real alerts in hazardous situations.

Active jamming consists of building an external radio transmitter that attempts to inject false signals into the radios. Thus, even if malware were to activate the radios and listen for position-revealing signals, it would, in theory, report largely bogus position information. This is particularly effective against GPS, where GPS signals are very weak and thus even a weak local transmitter should be able to overpower the GPS satellites. However, active jamming was ruled out for several reasons. The jammer's emissions could create a signal that can be traced to locate the reporter; the jammer will require substantial battery power, and the user is left vulnerable once the jammer's power is exhausted. Furthermore, nearby base stations may still be detected by the receivers, as modern radio protocols have sophisticated designs to protect against unintentional jamming.

Selective physical isolation or termination of the antennae consists of inserting an electronic switch between the connectors of the logic board and the antenna. The switch, when activated, would shunt the antenna to a matched resistive load, which would greatly reduce the transmission power and receive sensitivity of the radios. However, experimental verification on the WiFi subsystem indicated that removing the antenna connection and permanently terminating with a shunt resistor still leaked sufficient RF into the receivers for local base stations to be detected (e.g. within the same room), which could be sufficient information to betray a reporter's location.

Methods that Do Meet our Criteria

Upon determining that semi-intrusive countermeasures were inadequate, we investigated options that involve measuring signals on the phone's logic board, typically via test points designed in by the manufacturer. It is no surprise that complex systems such as the Apple iPhone 6 would have test points baked into the circuit board design to assist with debugging. These are an essential part of yield and customer experience improvement; defective units from the factory and the field are sent back to the headquarters, and engineers rely on these testpoints to determine the root cause of the device's failure.

Using repair manual documentation acquired from the Hua Qiang electronics market, we cataloged a set of internal test points that were:

1. Accessible with low probability of damage to the logic board by a trained operator
2. Could provide meaningful data on the radio status
3. Would be difficult or impossible to disable or spoof (e.g., future-proof against adversaries aware of our research).

For the accessibility criteria (1), test points were considered viable even if they required desoldering an RF shield or the SIM card connector, and manual removal of the soldermask. In our experience, a trained technician can perform these tasks with low probability of irreparable damage to the motherboard. These operations are not recommended for entry-level novices. However, our experiences in Shenzhen indicate that any technician with modest soldering skills can be trained to perform these operations reliably in about 1-2 days of practice on scrap motherboards. Thus, technicians could be trained to perform the modifications in any locale with sufficient demand for modified iPhones.

Table 1 summarizes the test points we have accessed and have found to provide introspection data that potentially meet criteria (2) and (3), and Figures 2-5 illustrate the location of the test points.

Signal name	Signal type	Signal function	Related radios
FE2DATA	Shared serial data bus	Configure antenna switches	Cellular radio antenna multiplexer, RX diversity, antenna tuning
FE2CLK	Reference clock		
FE1DATA	Shared serial data bus	Configure Power Amplifiers and Duplexers	Cellular radios
FE1CLK	Reference clock		
BBTX	UART	Baseband to AP comms	GPS, others
BBRX	UART		
WLAN RX	UART	WLAN to AP comms	WLAN
WLAN TX	UART		
WLAN_PERST	Reset	Reset PCI bus on WLAN	WLAN
BT RX	UART	Bluetooth to AP comms	Bluetooth
BT TX	UART		
GPS_SYNC	Sync status	GPS signal quality and sync	GPS

Table 1: Internal signal candidates for introspection.

Figure 2. The FE1, FE2 bus probe experiment. Test points from the back side of the PCB are wired to the top side for easy probing.

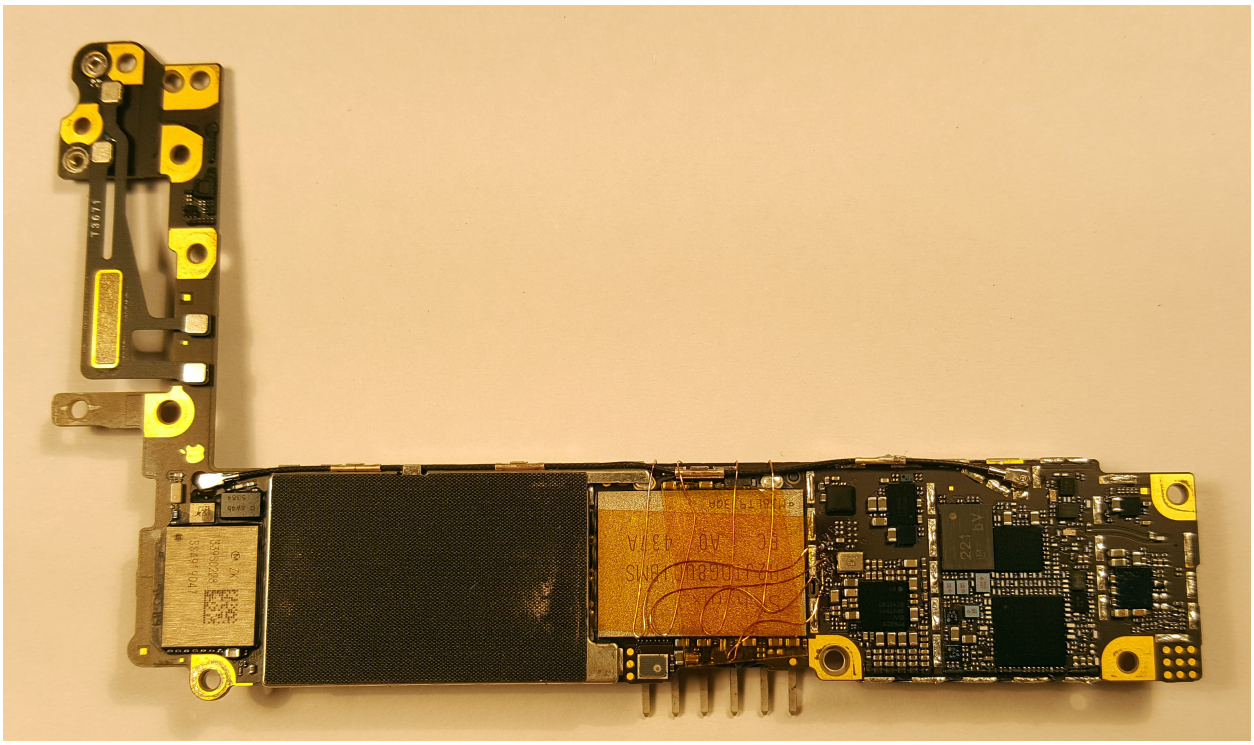


Figure 3. The backside of the FE1, FE2 probe experiment. The test points are located adjacent to the NAND Flash, underneath an RF shield which was removed for this experiment. The test points were covered with soldermask, which was removed through mechanical abrasion.

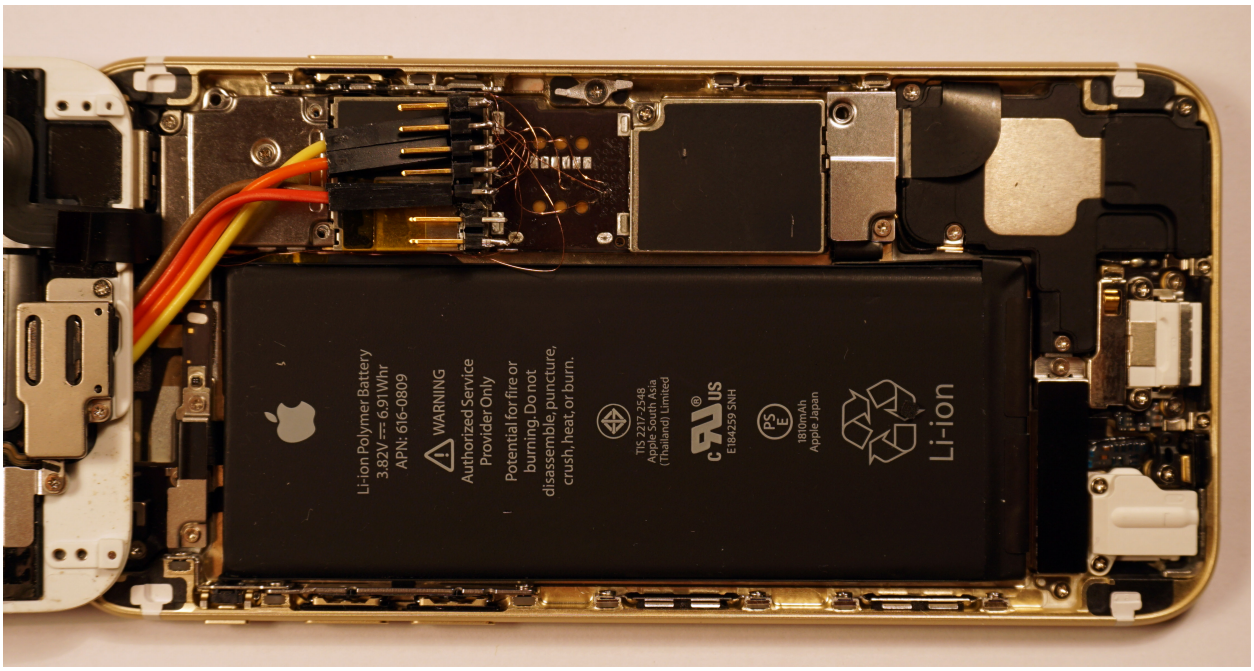


Figure 4. The UART and GPS sync probing experiment. The majority of the test points

are located underneath the SIM card connector, which was removed for this experiment.

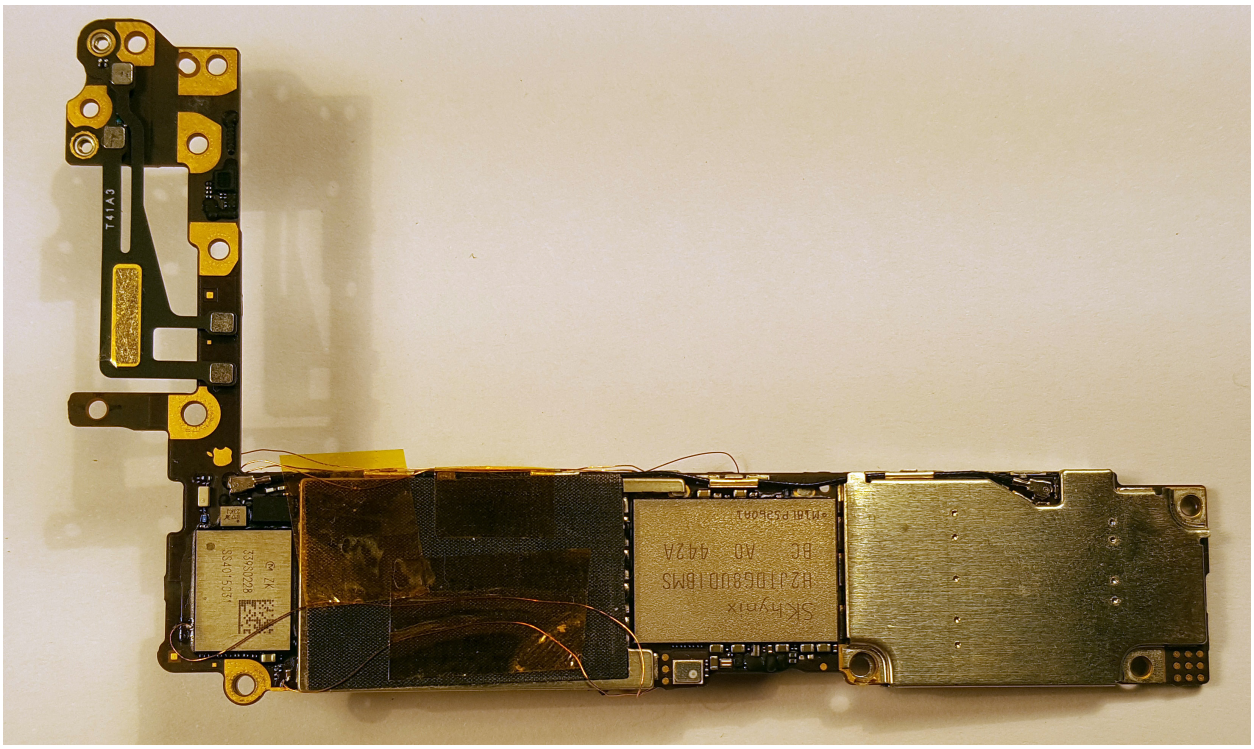


Figure 5. The back side of the UART and GPS sync probing experiment. A pair of wires are run to break out WLAN_PERST and power-related signals for monitoring.

Cellular Modem Introspection

The FE1 and FE2 serial buses run at 20MHz, with a 1.8V swing (Figure 6). This bus is used primarily to configure the cellular modem radios. When the radios are on, there is constant traffic on these buses. When in airplane mode, the traffic completely ceases. The serial buses appear to adhere to a protocol known as MIPI SPMISM (System Power Management Interface)[14].

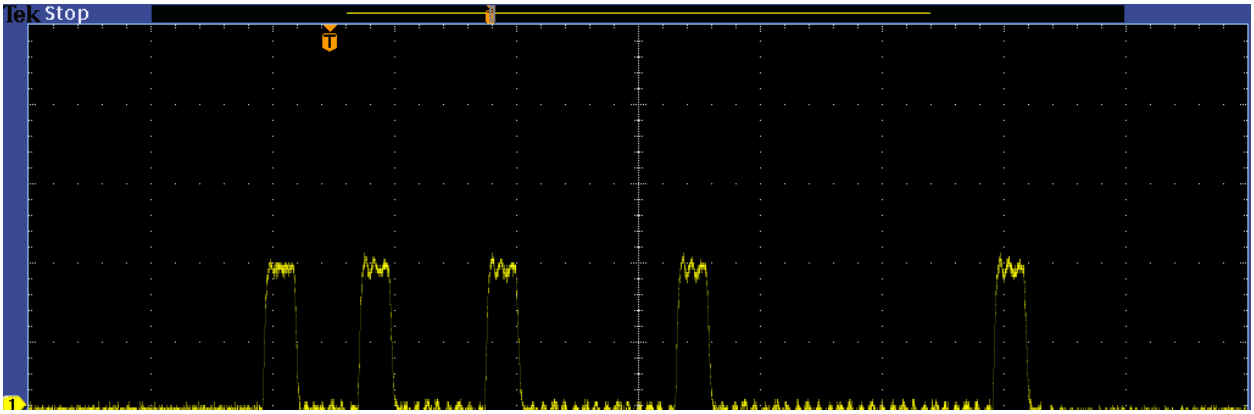


Figure 6. Example of bus traffic on the FE1 bus.

Cellular radios operate in a complex environment, and require constant adaptation of the antennae, power amplifiers, and band selection for proper operation. It is hypothesized that an attempt to even passively scan for base stations without transmitting will require traffic on this bus; at the very least, the antenna switches must be powered on and configured to receive. Therefore, cellular modem introspection may be as easy as noting if there is any activity on the FE buses during airplane mode.

We note for the sake of completeness that it may be possible for an attacker to statically configure the antenna, channel, and power amplifier settings and convert the device into a radio beacon that blasts out a signal that is inconsistent with the cellular modem standard but detectable through other means. In this mode, one would observe no traffic on the FE buses, but one could, in theory, triangulate the location of the transmitter with modified base stations or specially deployed receivers. This scenario can be mitigated by doing deep packet inspection and noting the addresses that should be accessed to power down the cellular modem systems. If any of the power-down addresses are skipped during the power-off sequence, that would be flagged as a potentially hazardous condition.

However, this scenario would require modifications to the cellular modem transport specifications, and as such one would need to deploy modified base stations across the territory to gain adequate surveillance coverage. This would likely require extensive cooperation of both the baseband radio vendors and cellular providers to craft and effectively deploy such an exploit. Because of the difficulty, we imagine such an exploit would be available only to well-organized government-level adversaries.

Finally, the phone’s vendor, Apple, could volunteer (or be coerced) to push a signed update that sends random “NOP” packets over the FE buses during airplane mode to force false positives and make this technique less effective. Again, in such a case, deep packet inspection could help to discard noise from signal. Although future hardware versions could encrypt this bus to foil observation, we believe it is not possible to introduce bus encryption with a software-only change: the peripheral devices on this bus lack loadable firmware. Thus, at least for current phone models, deep packet inspection should be robust.

WiFi & Bluetooth Introspection

The WiFi subsystem interfaces to the CPU through multiple buses, namely, PCI-express and a UART. The Bluetooth subsystem interfaces to the CPU through a UART, with a separate UART channel for coexistence. Because of the Bluetooth subsystem's relatively simple interface, it should be possible to robustly detect Bluetooth activity by simply monitoring the BT UART signals.

The WLAN UART signals seem to carry configuration and status information regarding WiFi configuration, as evidenced by the UART trace in Figure 7.

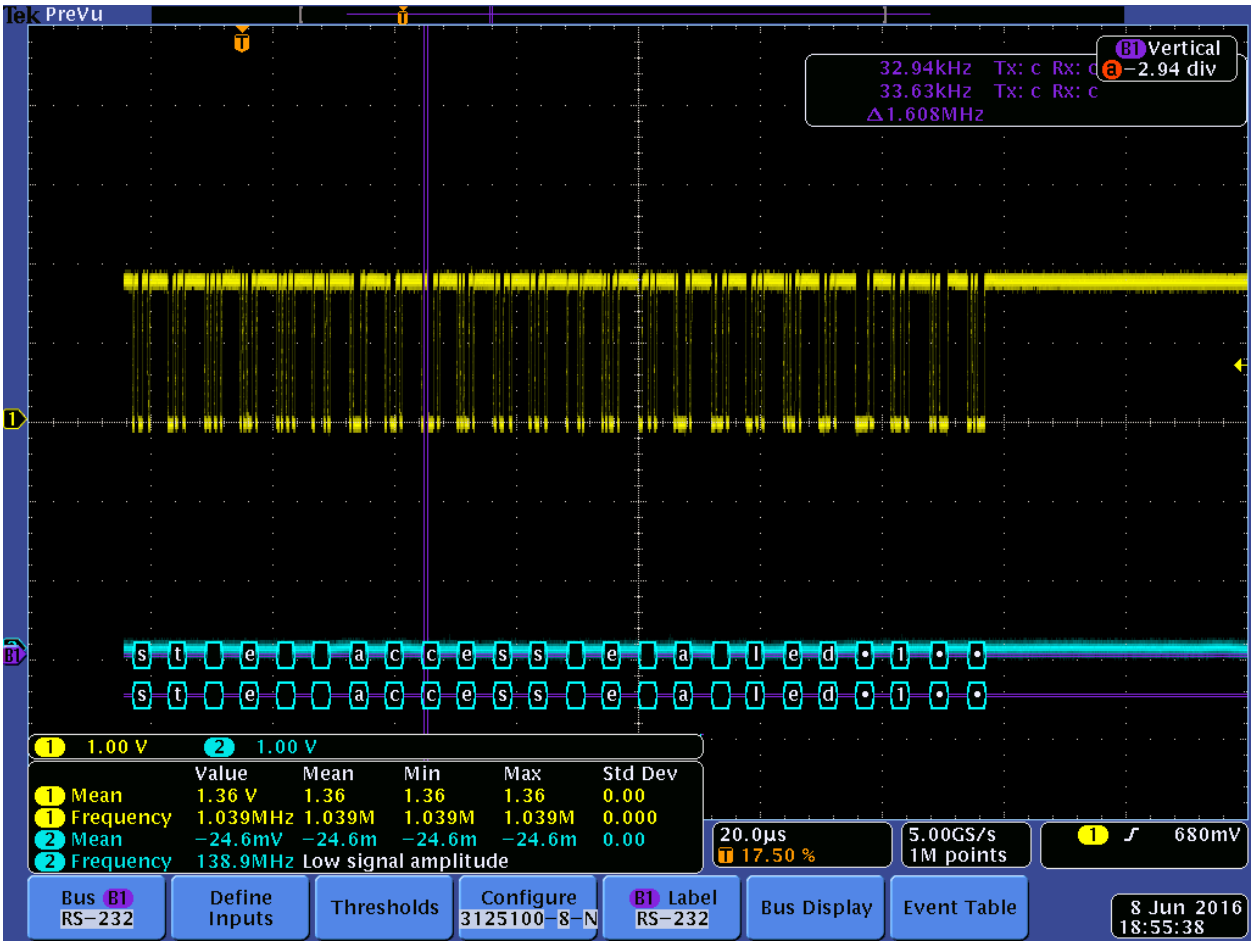


Figure 7. Example data on the Wifi UART as decoded by a Tek MDO4014B.

Further exploration of the data contained within the signals is necessary to determine if it is possible for an adversary to perform access point scans, which is an effective means of geolocation, without invoking the UART. Unfortunately, the WiFi power remains on even in airplane mode, so monitoring WiFi voltage levels has no correlation with radio activity.

Significantly, WLAN, BT, and GPS risks can be mitigated by forcing the WLAN PCI bus into reset. By holding WLAN_PERST low prior to power-on and throughout boot, WiFi will fail to enumerate on the PCI bus. iOS will continue to boot and is fully usable, but in the Settings panel, WiFi will appear to be off and cannot be switched on. Attempts to switch on Bluetooth fail, and GPS, although active, cannot access its antenna because the antenna for GPS is shared with WiFi. Note that forcing WLAN_PERST low during normal operation forces a phone reboot, so disabling WiFi using this technique effectively necessitates a reboot.

This is a simple but effective method to force several critical subsystems to be off, with no chance for an updated firmware to bypass a WiFi hardware reset. However, the failure of Bluetooth and GPS subsystems to activate may be due to firmware-only dependencies. It is hypothesized that these systems rely on WiFi to initialize before activating the respective antenna switches for these subsystems, since they all share a common antenna port. Thus it may be possible for an exploit to be developed to force Bluetooth and GPS to be on even if WiFi is in reset. Furthermore, it may be possible for malware to fingerprint systems where the WiFi has failed to initialize, and flag these users for further monitoring.

Thus, depending on the user's threat model, the WLAN_PERST defeat may be a simple but effective method to defeat several radios with a single signal, but it may also give away information to advanced adversaries on the presence of an Introspection Engine. Because of the effectiveness of the WLAN_PERST trick, we would present users with the option to activate this, but not require it.

Significantly, repair manuals indicate that the WiFi/Bluetooth module includes a hardware "RFKILL" pin. Apple leaves this pin unconnected and very difficult to access through mods, but if phone vendors wanted to support efforts like this, future revisions of phones could break such pins out to offer a more graceful defeat that doesn't require rebooting the phone or leave a measurable signature while disabling these radios.

GPS Introspection

To date, we have identified three possible methods for detecting GPS activation. One is to look for activity on the BB UART bus. When GPS is active, coordinate data seems to be transmitted over the BB UART bus. A second is to look at the GPS_SYNC signal. When GPS is active, the GPS_SYNC signal pings the base band at a rate of about once

per second, with a pulse width inversely proportional to the quality of the GPS lock. A very wide pulse indicates a high degree of uncertainty in the GPS signal. Finally, the GPS has an independent power regulator which is turned off when the GPS is not active, to save power.

NFC Introspection/Defeat

For NFC, we decided that the risk/reward of selectively enabling and monitoring Apple Pay is not worth it. In other words, we do not expect journalists operating in conflict zones to be relying on Apple Pay to get their work done. Therefore, to simplify the effort, we opt to fully disable Apple Pay by disconnecting the RF front end from its antenna.

Fortunately, the NFC's antenna is connected to the main logic board via a single screw. By removing this screw and separating the antenna from the main logic board, we hope to substantially and selectively reduce the sensitivity of the NFC radio. Further testing is required to determine if this is sufficient to guard against attacks by adversaries using high-power amplifiers to query the Apple Pay NFC feature. If found inadequate, further countermeasures, including but not limited to permanently removing the Apple Pay NFC RF front end chip from the mainboard, are options to prevent exploitation of the radio without leaving a clear signature that can be detected by an adversary.

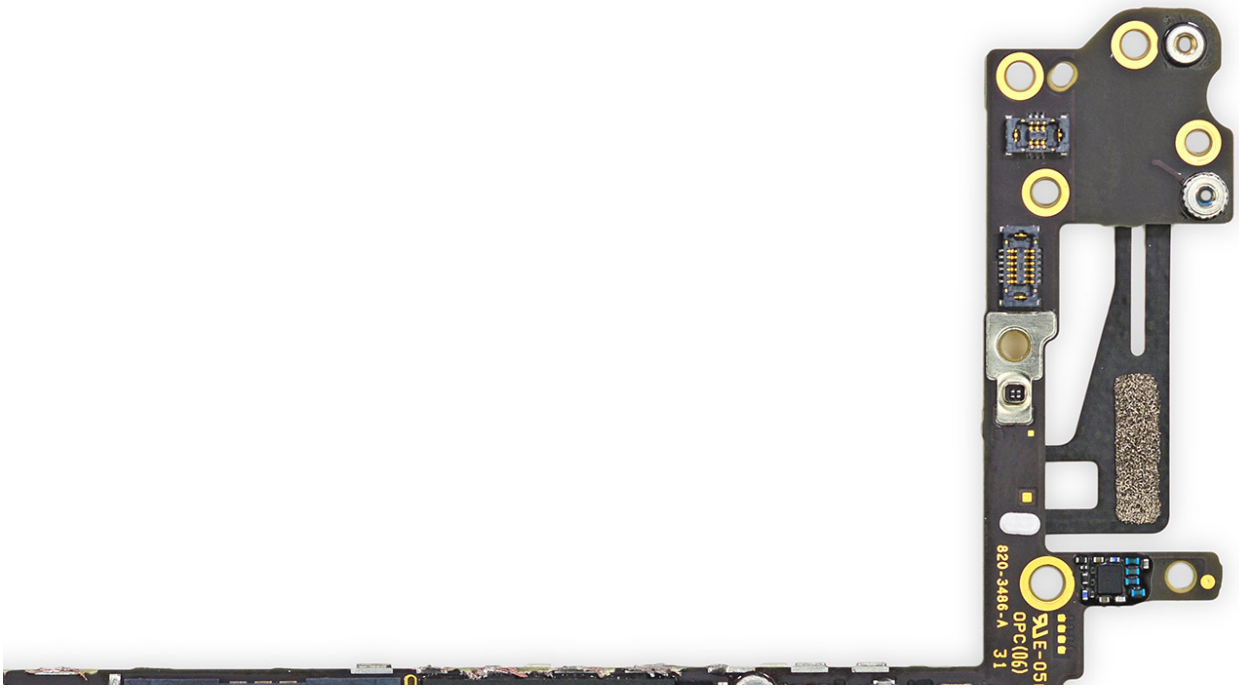


Figure 8. Location of the Apple Pay antenna connection, highlighted in pink. Original image courtesy iFixit, CC-BY-NC-SA licensed.

Long-Term Viability of PCB-Level Introspection

One criticism of direct introspection is that as mobile phone chipsets become increasingly integrated, PCB-level introspection will become difficult, if not impossible. There are already examples of this level of integration happening in low-end mobile phone chipsets, such as the MT6260 single-chip 2G AP+Baseband solution by MediaTek. From the outside, the chip appears as a regular BGA component, but X-ray imaging reveals it is in fact composed of several discrete pieces of silicon. In Figure 9, the thin, gracefully arcing darker lines in the X-ray are bond wires. Although the outline of the silicon chips is difficult to resolve in an X-ray, one can infer their perimeters by the pattern of bond wires that typically cluster along the edges of the chips. One can observe that the bond wires not only arc from the chips to the PCB, but also between chips. In such a design, one could imagine running control busses directly between chips over bond wires, making direct introspection very difficult.

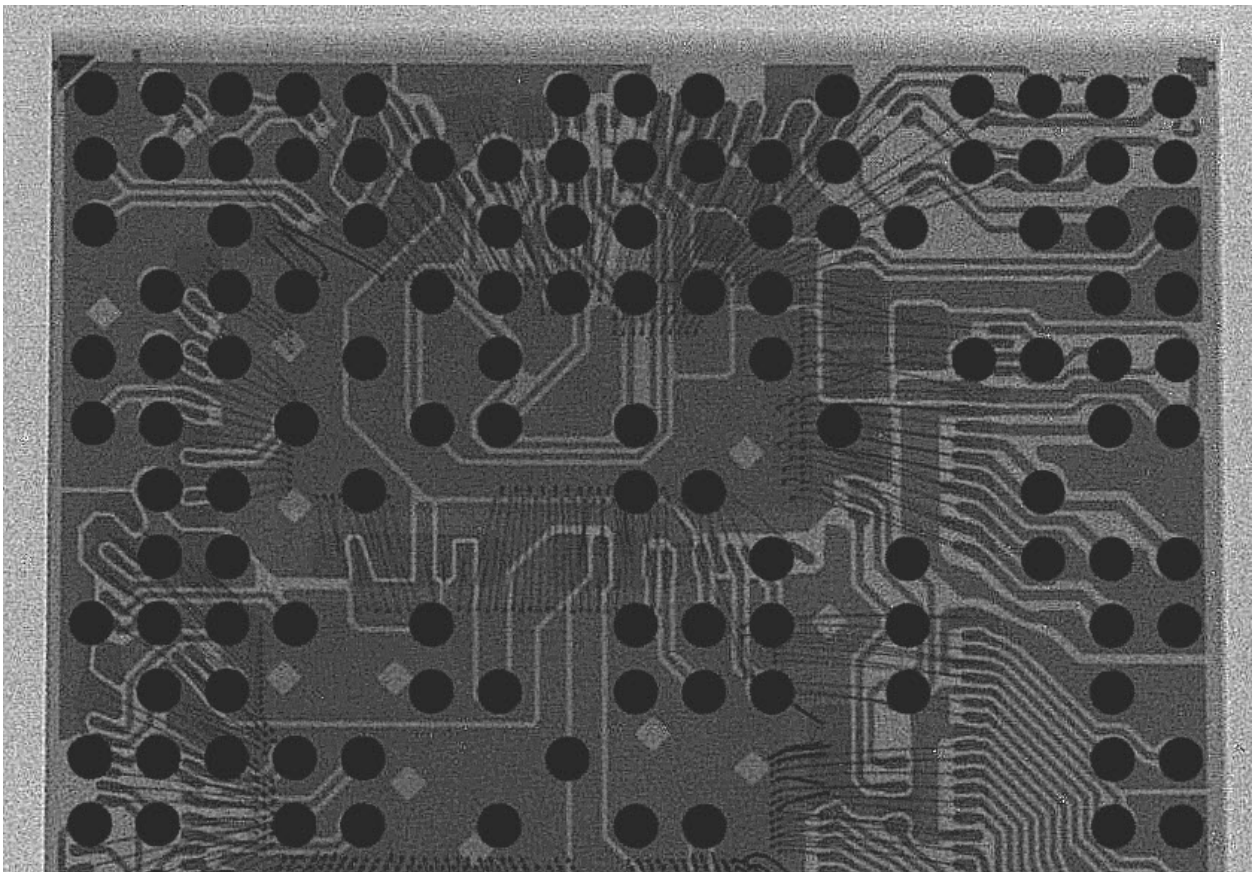


Figure 9. An X-ray of the MT6260, a single-chip 2G AP+Baseband solution by MediaTek. Image courtesy Nadya Peek, with permission.

Despite this extremely high level of integration, the RF power amplifier and LNA are still in a separate package and wired to the MT6260 with board-level control signals. A separate project, Fernvale[15], endeavored to reverse engineer the MT6260 hardware, so we know that, for example, the antenna TX/RX control switch, power amplifier enable signal, and band select control signals are laid bare for introspection.

Why did MediaTek stop short of integrating the RF electronics into the BGA package? The answer to this question is not as clear, but it is likely due to both physics and economics. From the physics perspective, thermal load, signal integrity, and crosstalk all argue against integration. Cellular power amplifiers may generate several watts of RF signal, which greatly increases the thermal burden of an integrated package. Furthermore, RF frequencies favor QFN-style packages, where the chips are mounted directly onto metal leadframes that serve dual roles as a heat sink and as a solid RF ground reference. This conflicts with the objectives achieved by a BGA package, namely high signal density and lower cost per pin. Furthermore, cellular receivers may have sensitivities better than -110dBm (~10 femtowatts), while GPS receivers need to be better than -125dBm (~300 attowatts), so package-level isolation of sensitive receivers from noisy digital circuits is desirable. From the economic perspective, the design of RF receivers and power amplifiers is still a specialty, and it may simply be uneconomical for MediaTek to negotiate the purchase of bare die from third-party vendors for package-level integration into their chipsets. Instead, it seems MediaTek had intentionally dis-integrated the RF front end and ensured the presence of multiple vendors to create a competitive market for such chips.

All this points to the likelihood that for some time, direct introspection will remain a viable technique, as its targets are buses that connect between the discrete RF front ends and the highly integrated chipsets driving them.

Guarding Against the Evil Maid

Although the details of techniques for making tamper-resistant or tamper-evident hardware are beyond the scope of this paper and well-documented elsewhere [16] [17], for the sake of completeness it is useful to have a brief discussion about the “Evil Maid” threat scenarios facing the Introspection Engine and possible methods of mitigation.

As the name implies, “Evil Maid” threats refer to a class of attacks on hardware where an adversary gains direct access to the hardware and tampers with it – perhaps reflashing the firmware, replacing circuit boards, or modifying the existing circuits. In the case of the Introspection Engine, the Evil Maid may manifest as anything from a literal maid who tampers with the phone while cleaning the premises, to a border inspection where the phone is examined in private within a state-operated facility, perhaps for an extended period of time.

Simple threats, such as JTAG reflashing of the MCU, can be guarded against by blowing the fuses on the MCU that prevent firmware upgrades or mass erasure; the MCU chosen for the Introspection Engine proof of concept implementation supports both options. Significantly, the Introspection Engine is explicitly not to be field-upgraded: field units should not support any simple firmware upgrade option to guard against trivial Evil Maid attacks.

The Introspection Engine is also designed for easy self-test, in the sense that one can verify that introspection is working by simply bringing the phone out of airplane mode and observing that all the monitored signals go live. If a monitored signal fails to report out of airplane mode, one can directly conclude that perhaps the Intropsection Engine is faulty or has been tampered with. Of course, such a test should be conducted only in safety, perhaps in advance of entering a danger zone.

The worst case scenario is a well-prepared, well-funded state-level adversary which prepares a custom version of the Introspection Engine’s MCU. This is within the capability of the US National Security Agency[18]. For example, the MAESTRO-II at 0.515” almost fits within the footprint of the ICE40-FPGA used in the current Introspection Engine prototype. Also, based on textual description, the JUNIORMINT would likely fit within the footprint of the smaller MCU package. These implants could be overmolded within a QFP (quad flat pack) leadframe and laser-etched with markings rendering them indistinguishable from genuine components, at least to the untrained eye. Finally, the relatively high power signature of such an implant could be masked by including a genuine MCU in the same package, and allowing the genuine MCU to run mock UI code thus conserving power until a trigger is detected which powers up the implant and executes the desired attack.

Such exploits could be mitigated by porting the Introspection Engine design from using easy-to-assemble QFP packages to using WLCSP (wafer level chip scale package) devices. QFP devices were chosen to enable technicians of moderate skill with simple tools to build an Introspection Engine from scratch. Although WLCSP

devices are more challenging for hand assembly, they are cheap and accessible thanks to their popularity in mobile phones. The advantage of WLCSP parts with respect to Evil Maid attacks is that they have no package – they are essentially naked pieces of silicon. Thus spoofing these would require fabricating custom silicon. Silicon fabrication is of course possible, but more difficult than overmolding an off–the-shelf implant module.

Most traditional static defenses in common use today such as locks, alarms, and tamper-evident seals can be overcome by a sophisticated and dedicated adversary given enough uninterrupted time-on-target. Based on the senior author's experience, even the US Central Intelligence Agency's covert field sites calculate the value of defensive measures not on whether they can be defeated, but rather how long those defeats are predicted to require of a skilled attacker. Still, those with a deeper interest in increasing such costs to mitigate the risk of covert entry threats are invited to review the Freedom of the Press Foundation & Guardian Project's Phoneypot research effort [19].

Similarly, we refer readers to discussions of PUFs – physically unclonable functions – to create tamper-evident seals and to mark circuit boards and key components[16]. A well-executed PUF can greatly complicated an attack even by a well-prepared adversary, creating a time barrier. Given that the Introspection Engine is meant to always be kept within field of view, defeating the tamper-evident seals can take several minutes, if not hours, for a well-trained adversary to bypass

The bottom line is that in reality, as a small-footprint security-critical device, the Introspection Engine is not meant to be left unattended. The general rule is if an adversary has unescorted physical access to the journalist's possessions, then the adversary wins; inserting a tracking beacon into the Introspection Engine is not the path of least resistance. For example, it would be easier to shim something into a shoe or a suitcase.

Implementation

Tapping the Test Points

So far, we have discussed the discovery and mapping of test points on the iPhone6 motherboard. Unfortunately, test points are scattered around the motherboard, and

are difficult to identify through casual inspection. In order to facilitate harnessing a phone with test points, we have developed a technique for creating a wrap-around flexible printed circuit (FPC) with tapping lands that coincide with the test point locations. We call this FPC the “tap board”.

Once the tap board is tacked in place using a set of easy-to-locate physical guides, the tapping lands within the tap board will tend to lay over their intended test points, easing the process of harnessing the phone while simultaneously providing a method for managing and routing the signals to a single FPC connector. Another advantage of the tap board is it modularizes the phone-specific tapping process, thus allowing the mix-and-match of various tap boards with various signal analysis modules.

There are three major challenges to address when creating the tap board:

1. Accuracy: Test points can be as small as 0.3mm on a grid as fine as 0.8mm, over distances approaching 100mm. This means measurements of test point locations must be accurate to 1 part in 1000.
2. Double-sided: The same accuracy has to be maintained while wrapping around to the opposing side of the motherboard.
3. Resilience: The tap board’s structure must be able to absorb macro-scale offsets due to expected process variations and manufacturing tolerances, without disrupting the relative accuracy of test points.

The first step in creating the FPC is to create a high-resolution, scale-accurate scan of the motherboard. As modern CMOS flatbed scanners have a very shallow depth of field, one may need to first remove any tall components from the motherboard. This means that creating a tap board design will require the sacrifice of a phone. Fortunately, for the iPhone 6, there is a liquid market for scrap material at all stages of production, and we were able to easily secure a blank motherboard which was scrapped very early in production due to subtle internal defects within the PCB (Figure 10).

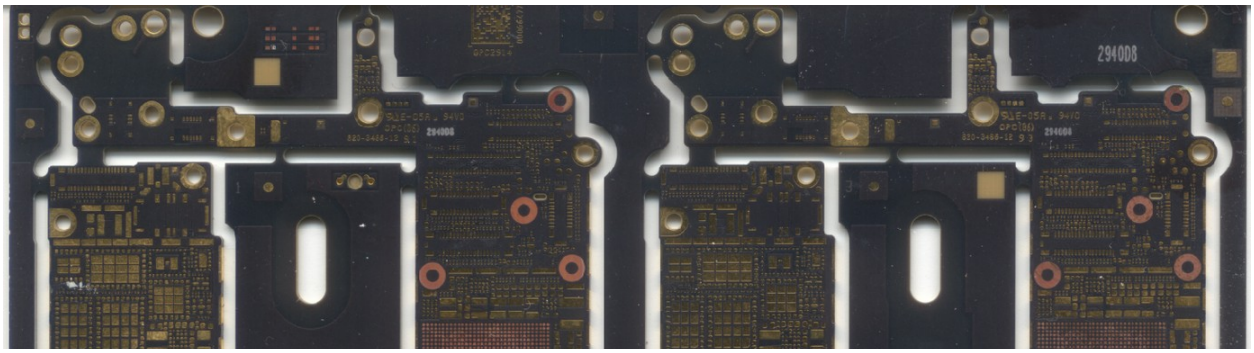


Figure 10. A scan of a blank iPhone6 motherboard panel. This panel contains four copies of the motherboard.

In our case, we captured a scan of the motherboard at 1200ppi using a CanoScan D660U. Once captured, the rotational accuracy of the panel was corrected to the pixel level, and the X and Y scales independently confirmed using a digital caliper to an accuracy of 0.02mm. Due to the imaging mechanism of flatbed scanners, is important to correct for the X and Y scales independently, as they are not mechanically linked.

Once captured and corrected, the images were laid into a vector drawing program, such as Adobe Illustrator. Rough guide lines are drawn around the edge of the PCB, mounting holes located, and test points circled (Figure 11). While not strictly necessary, drawing in edges and mounting holes help serve as a sanity check on scale and alignment.

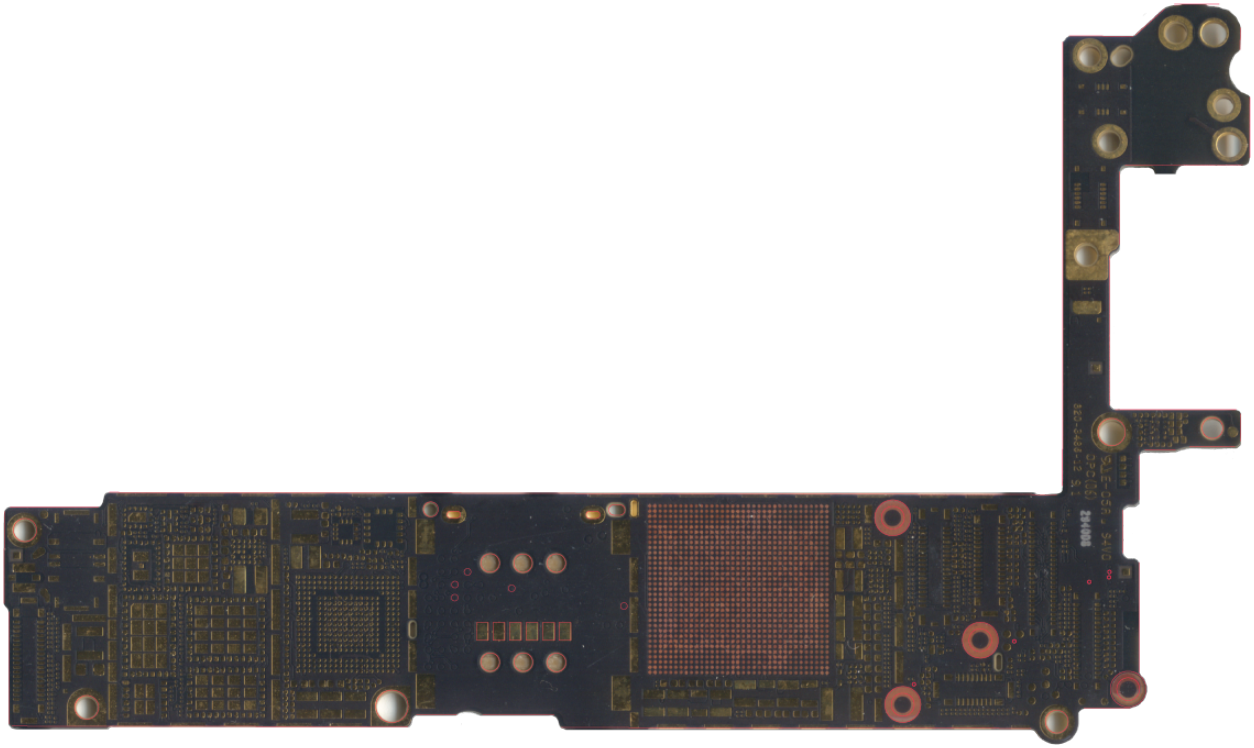


Figure 11A. Top side of the iPhone6 motherboard, rotation and scale-adjusted, cropped, and annotated with test points.

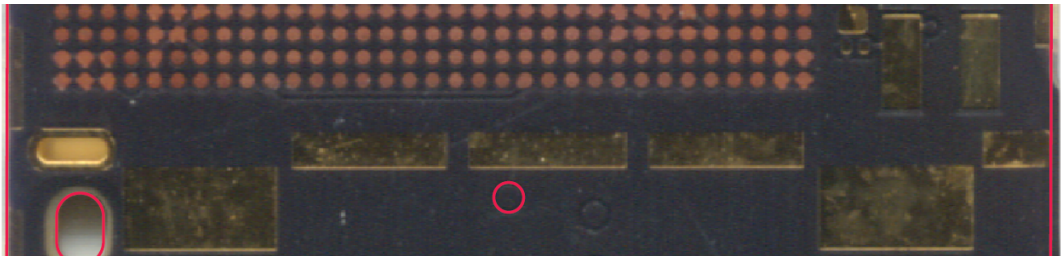


Figure 11B. Zoomed in view of the SIM card region, showing detail of the vector art annotation.

Once annotated, the Illustrator file was exported to a DXF format, and subsequently imported into Altium Designer (Figure 12). These imported outlines form the basis for creating the layout of a scale-accurate tap board.

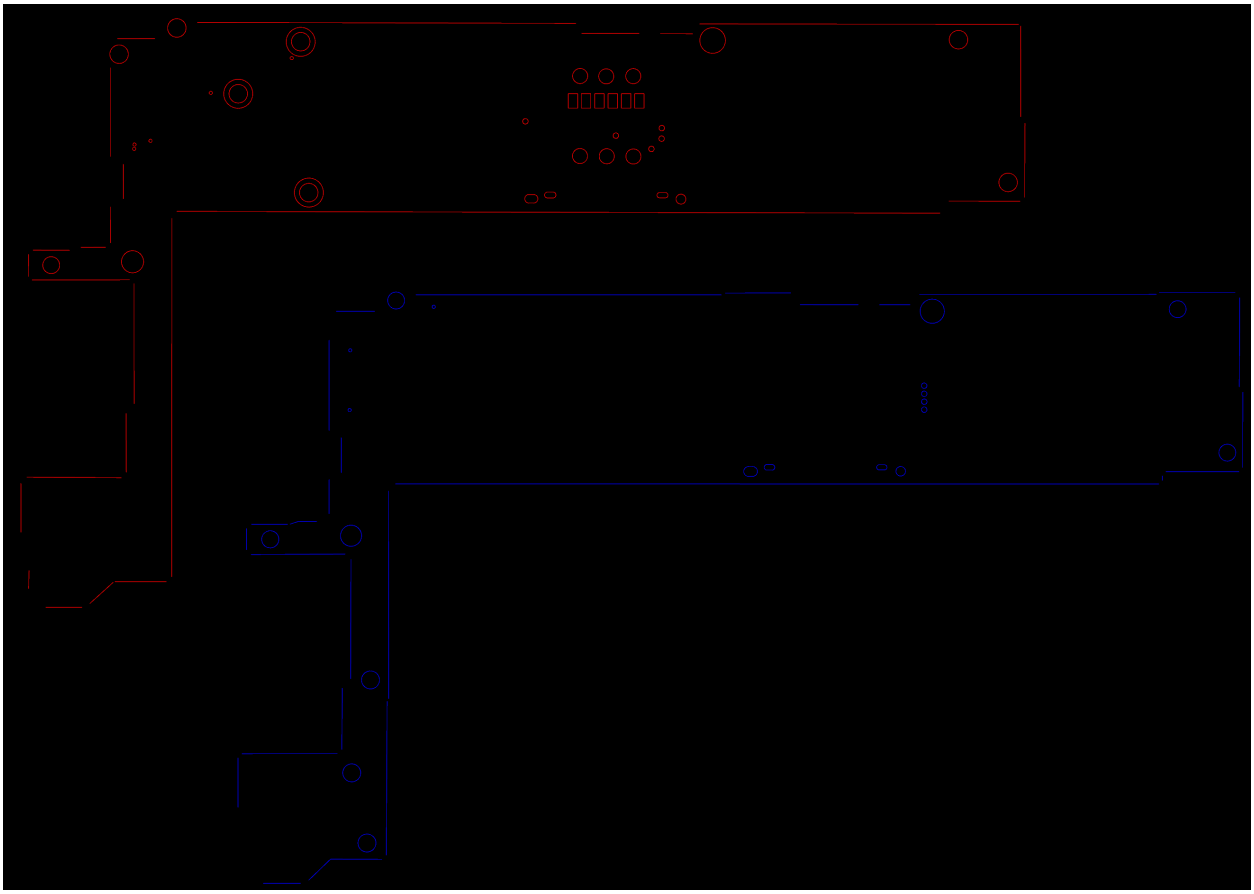


Figure 12. Initial import of basic outlines and test points into Altium Designer.

The outlines are then rotated, flipped and spaced, so that front and back side are facing the same surface of the FPC design. The spacing between the front and back side reflects the anticipated “Z” height that must be spanned by the tap board FPC.

Up to this point, we have addressed challenges 1 and 2 of creating the tap board, namely, scale accuracy and maintaining this accuracy over a double-sided layout. The final challenge, resilience, is addressed during the routing of the test signals within the FPC. Long runs of signals are routed along a serpentine pattern, allowing the FPC to stretch and absorb large-scale tolerance issues induced by variations in component heights and manufacturing tolerances. These techniques can be seen in the final layout of the tap board in Figure 13.

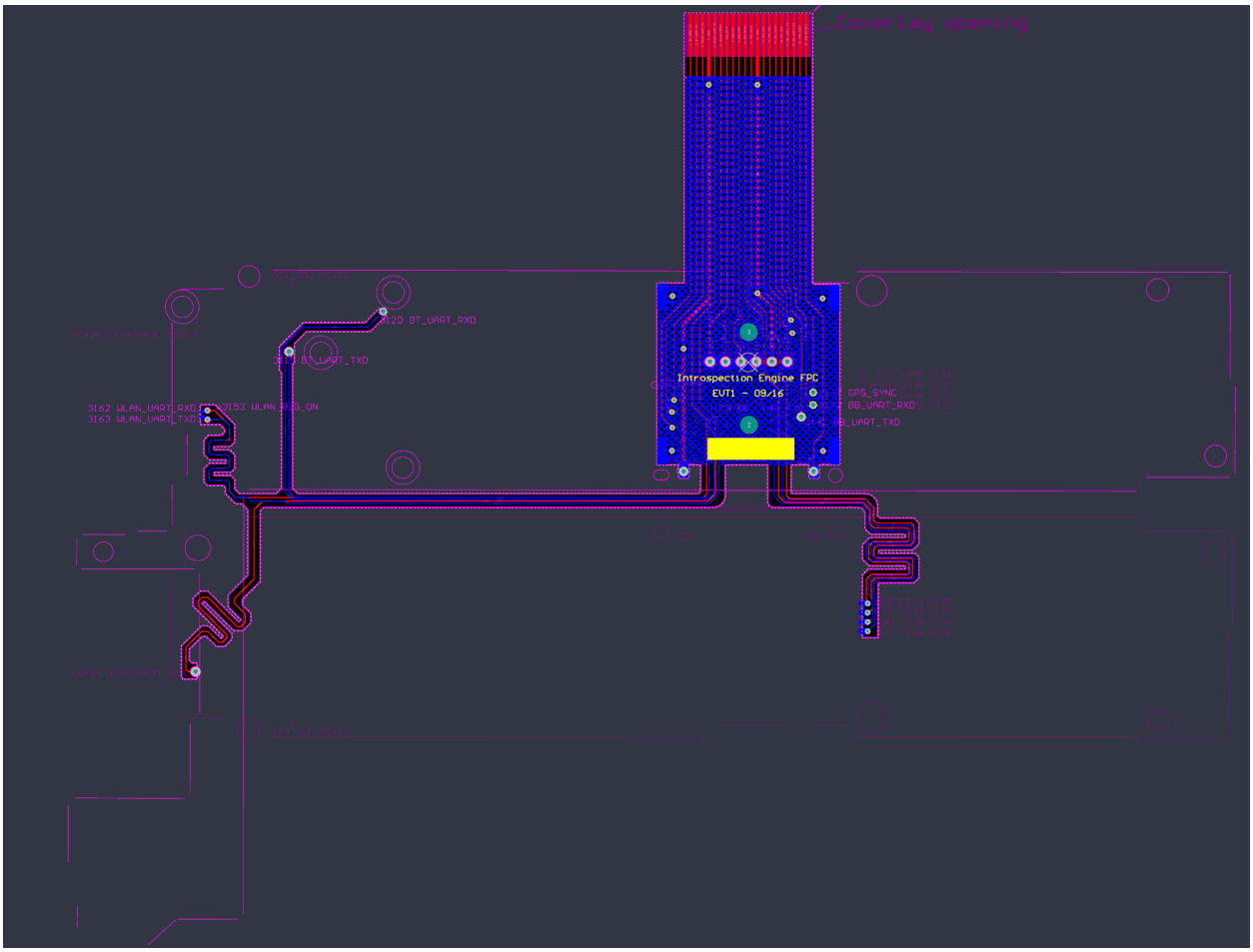


Figure 13. Final layout of the tap board FPC in Altium Designer[20].

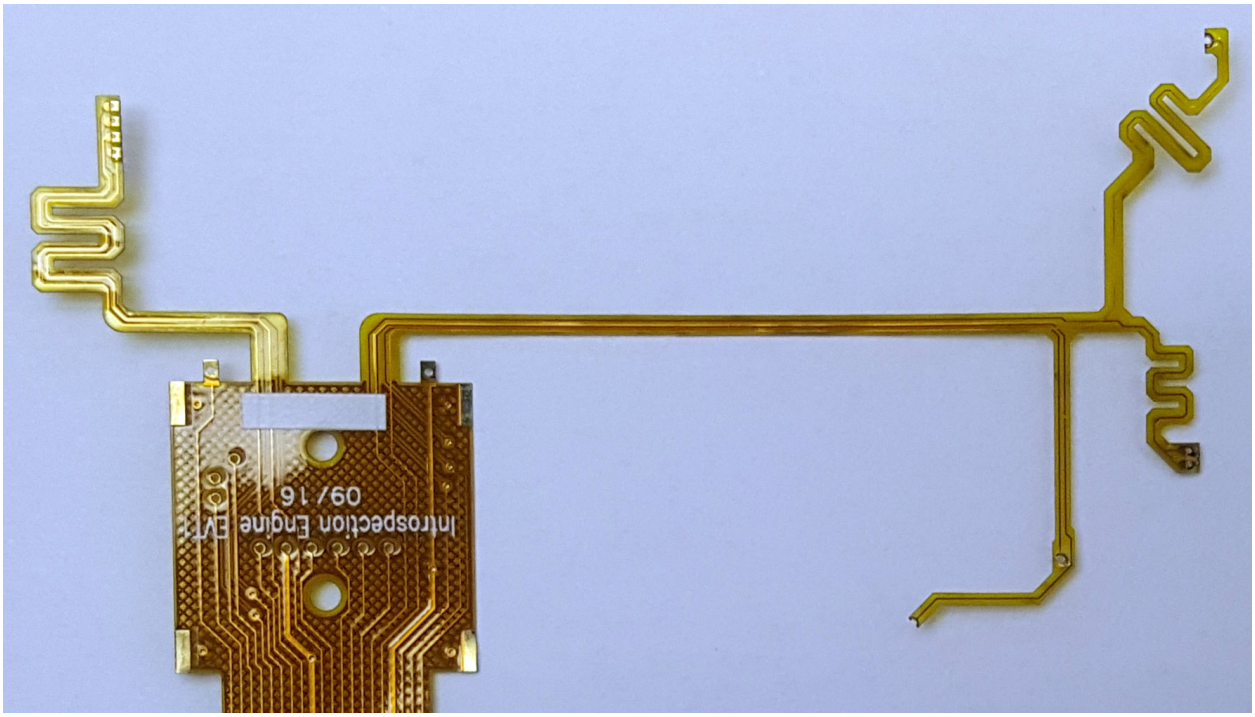
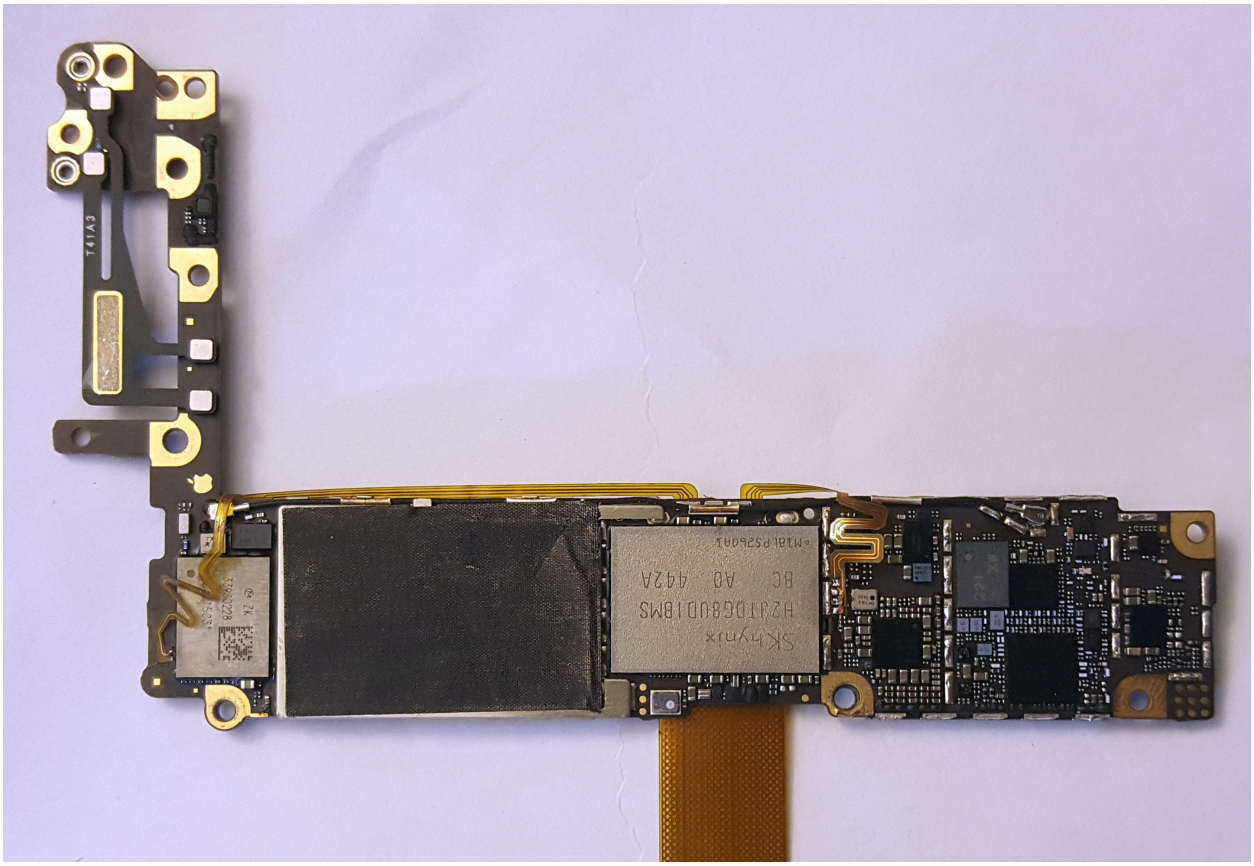


Figure 14. Tap board as fabricated.

Once fabricated (Figure 14), the tap board must be installed within the iPhone6. To do this, one must first remove two components from the iPhone6 motherboard: 1) the SIM card slot and 2) the RF shield covering the lower bottom portion of the PCB. Removal of both of these items requires access to a hot air soldering station, and is considered a routine operation for phone repair technicians. These components must be removed because they obstruct access to key test points. The RF shield is not necessary for the phone to function and can be left off after harnessing. The SIM card slot is restored by routing the SIM signals on the tap board connector to a pair of card slots located on the prototype Introspection Engine signal analysis module.

Attaching the tap board requires scraping back the soldermask covering the test points, laying the tap board over the test point, and blobbing solder onto the assembly. The tap board has via holes strategically located over the test points so that the solder can flow through the tap board and directly heat up the test points, easing the harnessing operation while reducing chances of accidental short circuits. Figures 15 and 16 illustrate the tap board installation, both on the naked motherboard and in the context of the entire phone.



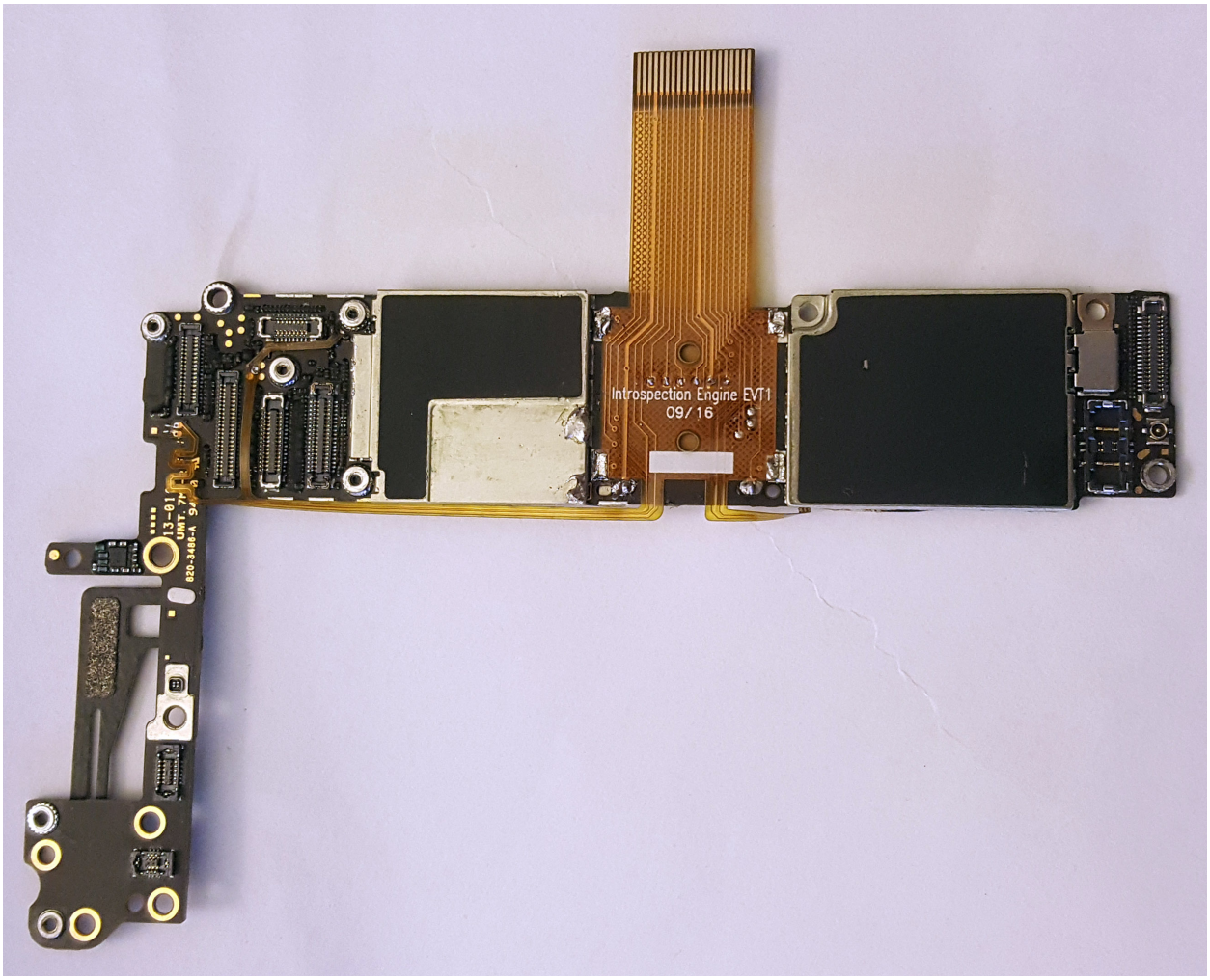


Figure 15. Tap board as installed on an iPhone6 motherboard. (Upper) Front view. (Lower) Back view.

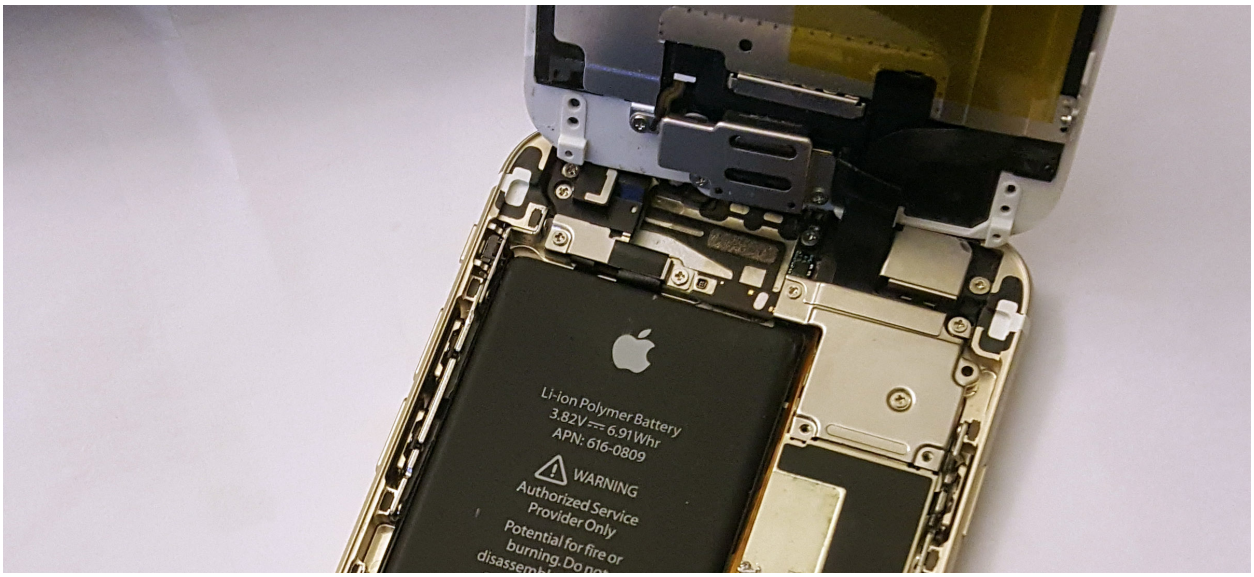


Figure 16. Views of the installed tap board within the greater context of an iPhone6. (Upper) Open iPhone. (Lower) Closed iPhone.

Signal Analysis Module

As alluded to previously, the Introspection Engine is broken into two major parts: the tap board and the signal analysis module. By breaking out mission-critical introspection signals with a tap board to a common 0.5mm pitch FPC connector format, users can mix-and-match phones and analysis modules. Being able to swap out analysis modules means we can avoid building a complex, one-size-fits all analysis module which inevitably leads to challenges in validation, may necessitate firmware updates, and present an overall inflated attack surface. Instead, we can build targeted, minimum viable modules which are easier to inspect, maintain, and secure, with each module customized for a given set of threat scenarios.

For this proof of concept research, we developed a simple signal analysis module which is capable of counting events on the critical introspection buses: SPMI, UART, and GPS. Event counting is analogous to counting network packets: one knows traffic has happened, but nothing about the nature of the traffic. Event counting was chosen under the theory that in airplane mode, no packets should be sent at all, therefore a near-binary indicator of traffic is sufficient. One could chose to implement a signal analysis module which can log and inspect the radio bus traffic using more sophisticated filters, but it would require substantially more capable hardware to

keep up with the relatively high bitrates present on these buses (20 Mbps for 2x SPMI and 3 Mbps for 3x UARTs).

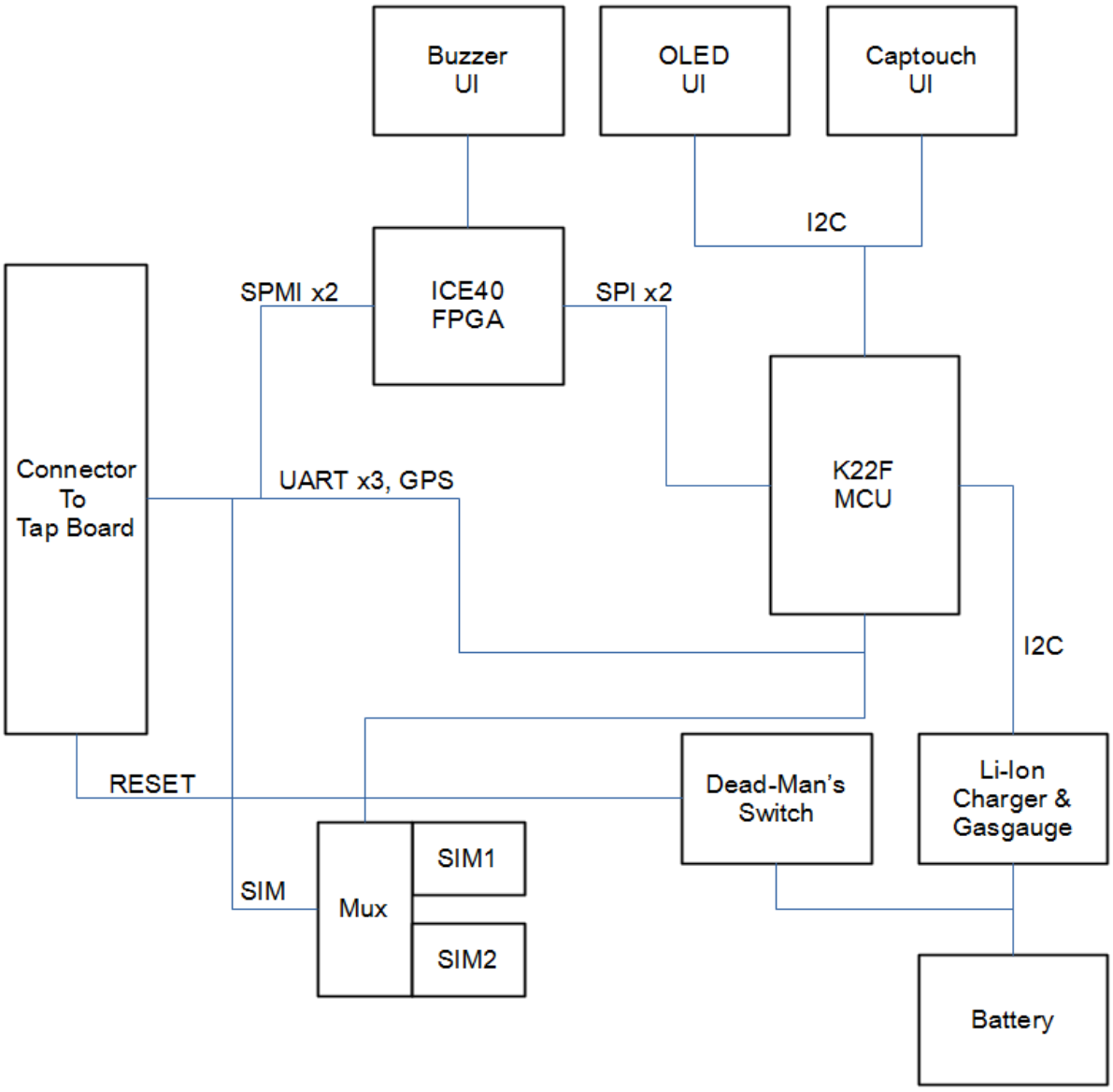


Figure 17. Block diagram of the proof of concept signal analysis module.

Thus, the design goals for the proof of concept signal analysis module are as follows:

- Ability to count and log packet events on the relevant buses
- Simple hardware design using the most open and inspectable components available at the time of design
- Relatively simple code base, allowing for easy audit and verification

The block diagram in Figure 17 outlines the basic architecture of the Introspection Engine's proof of concept signal analysis module. The schematic and layout implementation of the proof of concept module is openly available [22].

SPMI is not an interface commonly available on low-end microcontrollers; therefore, we opted to use a simple FPGA to assist with packet counting and logging. We chose to use the ICE40 FPGA from Lattice in part because it is one of the few FPGAs that has a 100% open source toolchain: Yosys for synthesis, Arachne for place and route, and IceStorm for bitstream manipulation and timing extraction [23]. A simple SPMI-to-SPI packet converter with FIFO was implemented in Verilog [24].

UI functions are controlled by a Kinetis K22F-series (MK22FN128VLH10) microcontroller. The chosen part contains a 100MHz Cortex-M4 CPU, 128K of FLASH, 24k of RAM, sufficient SPI, I2C, and UART resources to simultaneously present a UI as well as perform basic monitoring of the UART and GPS introspection taps. The K22F runs ChibiOS, an open-source real time operating system which is extremely compact and easy to analyze, despite offering features such as a HAL, multi-threading, and synchronization primitives [25].

In addition to the analysis functions, the signal analysis module includes two SIM card slots, which are connected to the single native slot on the iPhone6 via an analog multiplexer; the SIM card detection signal is controlled via the K22F to simulate a card unplug/plug event when swapping between SIM cards.

Finally, the analysis module contains an integral lithium-ion battery and corresponding charger/gas gauge circuit, along with a "dead man's switch". The purpose of the dead man's switch is to force the iPhone6 into a safe state when the battery dies on the Introspection Engine. We accomplish this by using a depletion mode FET across the reset line to the iPhone6. A depletion mode FET is normally conducting in its unbiased state, so if the battery is removed or is exhausted, the bias circuit fails, and the iPhone6 is forced into reset. Although conceptually simple, such a circuit is difficult to execute because by definition it needs to operate under a wide range of marginal conditions. The proof of concept circuit here works for simple cases but still needs improvement. For example, reset of the iPhone was meant to be tripped when the low voltage monitor built into the K22F trips the system reset. Although the documentation of the K22F claims this feature exists, lab tests show it is unreliable. Thus, a separate, discrete low voltage monitor circuit should be added to ensure reliable reset of the iPhone. Note that the depletion mode FET is still necessary

despite the low voltage monitor, because the low voltage monitor circuit still requires a non-zero voltage to operate properly.

Initial Results

To finalize the proof of concept, we fabricated both the tap board and the signal analysis module, and installed them on target devices. The tap board was installed on two phones, once by the author, and once by an experienced technician who had no specific prior briefing on how to install the tap board, in order to validate that the tap board does in fact simplify the harnessing process. The technician was able to install the board successfully in about an hour of effort on the first try.

The signal analysis module was fabricated and installed in an off-the-shelf battery case for the iPhone6 [26] that was gutted and lightly modified to fit the signal analysis module electronics. The resulting assembly can be seen in Figure 18 and in Figure 19.

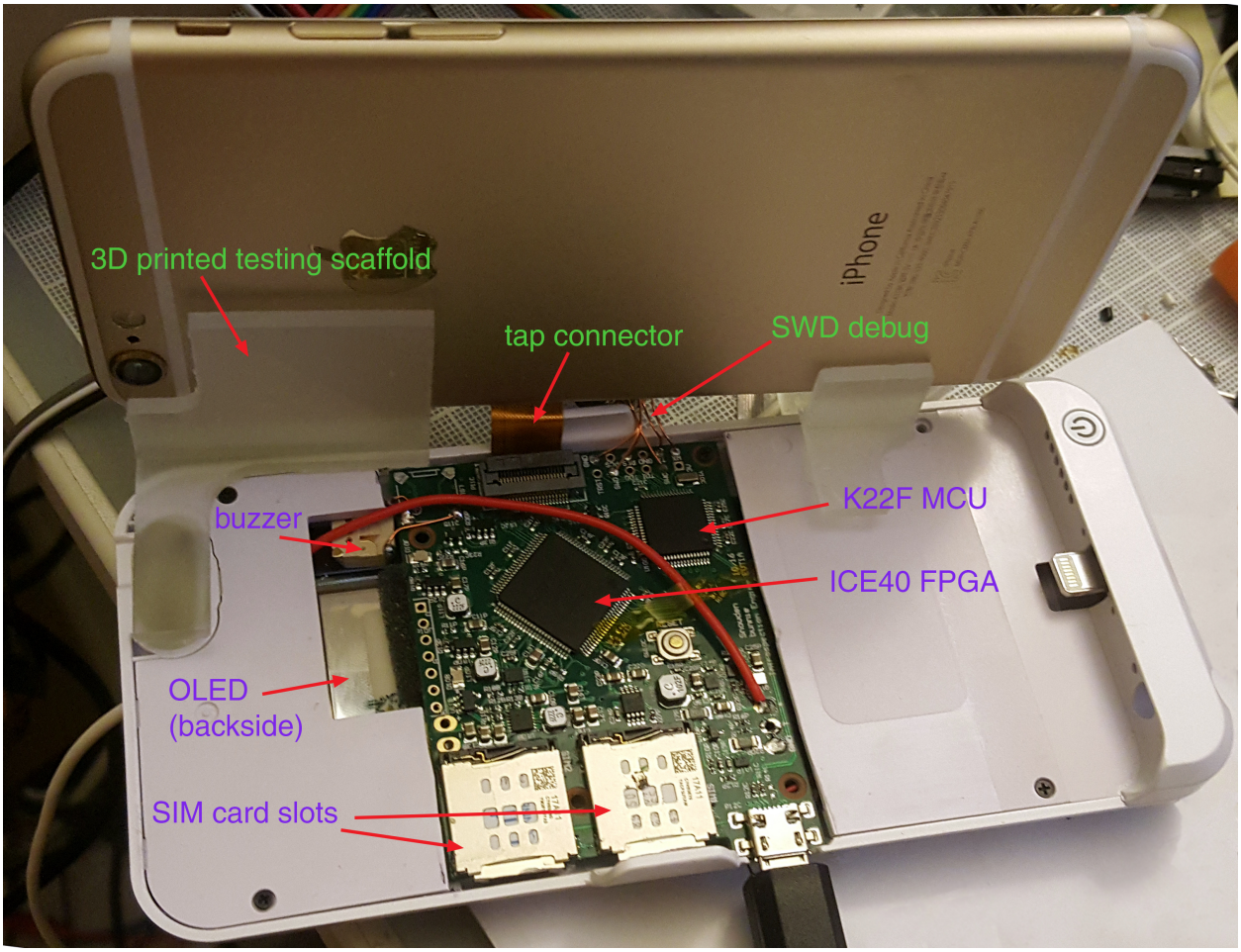


Figure 18. Signal analysis module implementation and mounting in a modified off-the-

shelf battery case.

Figure 19. iPhone 6 mated with the Introspection Engine prototype.

A simple user interface was crafted to track and manage stream of data coming into the Introspection Engine. In this proof of concept, users are presented with a graph over time of activity on the four monitored radio buses (Figure 20). Users can optionally set audible alarms and notifications based upon observed transitions in and out of airplane mode, as well as select the active SIM card.

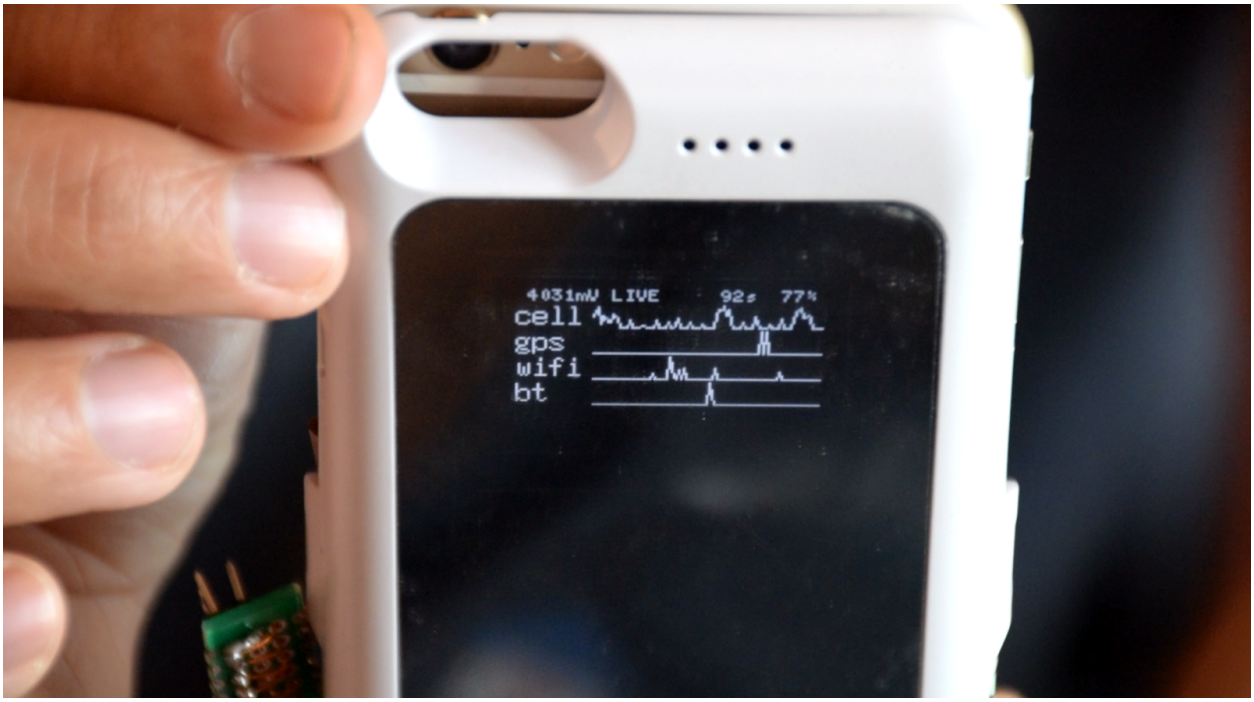


Figure 20. Example of the UI on the Introspection Engine, showing a graph of historical event frequencies on various radios.

Preliminary results using the Introspection Engine are encouraging. It successfully monitors and determines when radios are active. However, there are some confounding factors. For example, when the device is put into airplane mode, we occasionally see small amounts of traffic on WiFi and GPS – perhaps once every five minutes or so, some kind of communication occurs with the WiFi module. Furthermore, whenever a user first transitions a phone out of standby mode into a screen-on mode, for example by hitting the home button, there is often a short burst of activity, accompanied by a glitch on the GPS signal which we interpret to be caused by the GPS unit transitioning power states. These confounding factors are a violation of principle #4 – namely, avoiding false positives that can condition a user to ignore or turn off alarms.

There may be completely benign explanations for these – perhaps some routine houskeeping functions are taking place, or perhaps the phone is occasionally waking up and scanning WiFi, even in airplane mode, to accelerate AP discovery and re-association when taken into a connected mode. However, the current event-counting implementation doesn't provide enough data to analyze the WiFi UART traffic.

Thus, one possible future direction would be to modify the firmware to provide detailed logs of the WiFi UART traffic so one can attempt a deeper analysis of the traffic and thus differentiate between routine housekeeping traffic and more nefarious traffic.

Another possible direction could be to include a discrimination threshold, so the extremely sporadic "normal" traffic patterns seen in airplane mode don't trip the alarm, but are reported with a confidence level indicating how risky these events may be. While this resolves the conflict with principle #4, it does open up the possibility that an attacker could mask nefarious traffic patterns to look like housekeeping traffic in airplane mode. However, the update rate would be limited to only a couple events every few minutes, which would severely limit the rate of data leakage. The phone might be able to slowly scan and accumulate sufficient data to determine an accurate position overnight, but any attempt to relay this data to a remote adversary would probably be detected.

From Introspection Engine to Silent Phone

We have prototyped and verified the Introspection Engine's ability to tap and monitor critical radio signals within an iPhone6. Using the Introspection Engine, we were able to derive hardware-level information previously unavailable about a device's radio state. The generalized technique of identifying radio control buses and observing them in real time is a potent analytical tool for those concerned with detecting surveillance malware.

However, the discovery that events may happen on some of the radio buses during airplane mode prompted us to consider what if iOS is in fact turning on wifi during airplane mode and performing Access Point scans to, for example, help build out a database of MAC addresses versus location to improve Apple's mapping service? It would certainly be within their right to do so if users agreed to Apples' terms of use. In this case, we could not rely upon the Introspection Engine to protect journalists if legitimate system vendor code routinely powered up the radios to perform location scans.

In this scenario, the mission of protecting journalists would be better served by a "Silent Phone" – a phone or iPod-style device which has its wifi, GPS, bluetooth and baseband radios permanently or selectively disabled through a hardware defeat. Such a phone would then be convincingly air-gapped, having taken the hardware equivalent of a reliable vow of silence, and upon access to a safe location the reporter can plug in a wired Ethernet adapter via the charging/data connector built into the phone. Such a Silent Phone would give reporters the necessary tool they need in the field to get journalistic work done, while ensuring their safety by only allowing communications to occur only after taking the conscious step of physically connecting additional, single-purpose hardware.

Fortunately, in the course of developing and testing the Introspection Engine, we serendipitously discovered that iOS boots gracefully even when the baseband and the wifi/bluetooth/GPS subsystems are disabled via hardware resets. This property means that one can turn a regular iPhone into a "Silent Phone" by hitting just two test points, instead of the nearly dozen test points required for direct introspection. Furthermore, we believe the Silent Phone configuration would be extremely robust against any attempt to access the radios, since the respective chips would be held in hardware reset and unable to boot or initialize.

If the Introspection Engine were the equivalent of connecting a patient to a medical device that sounds an alarm at the outbreak of an infection at the site of an open wound, the Silent Phone would be the equivalent of amputating the limb on which that wound sat: certainly effective, but also drastic. However, thanks to the ability to plug in an Ethernet adapter, we can provide the user with a serviceable prosthetic for connectivity when it is required. Just as any medical system requires extensive clinical testing before administration in a life-critical situation, the Introspection Engine requires significantly more validation before one would use it in true a life-or-death situation. However, a Silent Phone approach, like an amputation, doesn't require FDA approval – and it is a treatment that can be performed with relatively crude tools.

We note that it may be possible to route the WiFi and baseband reset signals on to a physical toggle switch, which when flipped could selectively enable or disable the radios after a reboot. This might be desirable for users who are in less threatening situations where an accidental flip of the switch does not enable the targeting of artillery shells. However, for the journalists' use case, we felt that a reliably silent phone would have greater mission value than a phone which could accidentally pocket-dial adversaries.

Significantly, we had attempted to replicate our findings on a Nexus 5X device. While the Nexus 5X booted successfully with the wifi subsystem powered down, it was unable to boot with the baseband modem powered down. This type of behavior is less surprising, given the amount of system-critical functions typically delegated to the baseband subsystems, and it hints that creating a Silent Phone out of a Nexus device could be much more difficult. We suspect iOS may be an outlier for being able to boot using stock firmware despite the hardware failure of multiple modems, but additional research would be required to confirm this finding.

We hypothesize that a Silent Phone communicating via Ethernet adapter to a single-board computer, such as a Raspberry Pi, configured to function as a router and strict firewall permitting only communications via a Tor bridge [27] over obfuscated communications protocols [28] could provide a user reliable access to the modern app ecosystem on a smartphone without the hardware ever becoming aware of its location. Additional research is required to confirm the hypothesis, but we believe that as long as the journalist herself does not intentionally supply the phone with information about its location (such as in files or chats stored on the phone regarding her future plans and intentions), in this configuration geodata is simply not available to be stolen. This property also creates interesting opportunities for denial and

deception against those seeking to inappropriately access the journalist's phone. For example, files containing misleading but convincing location data could be stored on the phone as a sort of honeypot, to be discovered by an abuse of so-called "lawful hacking" by authorities. Such plausible-but-false information would cause actions such as raids to occur at a false address, thus alerting the journalist that their device has been compromised and is being actively targeted for retaliation.

Given the relative simplicity, robustness, and elegance of the Silent Phone solution, we intend to pivot our efforts from validating the Introspection Engine to creating a set of Silent Phones and associated wired connectivity accessories for field use by journalists.

Citations

[1] Committee to Protect Journalists (June 19, 2017). Journalists Killed since 1992/Motive Confirmed. Retrieved from https://cpj.org/killed/

[2] Al Jazeera. "The Spy Merchants" investigation. Retrieved from http://www.aljazeera.com/investigations/spy-merchants.html

[3] Schneier, B. (2016) Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World. W. W. Norton & Company.

[4] Lipp, K. (October 25, 2016) AT&T Is Spying on Americans for Profit. Retrieved from https://www.thedailybeast.com/atandt-is-spying-on-americans-for-profit

[5] Stanley, J (May 2010). The Crisis in Fourth Amendment Jurisprudence. American Constitution Society. Retrieved from https://www.acslaw.org/files/ACS Issue Brief - Stanley 4th Amendment.pdf

[6] Meade, A. (April 14, 2016) Federal Police Admit Seeking Access to Reporter's Metadata without Warrant. Retrieved from https://www.theguardian.com/world/2016/apr/14/federal-police-admit-seeking-access-to-reporters-metadata-without-warrant

[7] Woods, A. (October 31, 2016) Montreal Police Spied on La Presse Journalist Patrick Legacé. Retrieved from https://www.thestar.com/news/canada/2016/10/31/montreal-police-spied-on-la-presse-journalist.html

[8] Knaus, C. (April 28, 2017) Federal Police Admit to Accessing Journalist's Metadata without a Warrant. (Case update) Retrieved from https://www.theguardian.com/australia-news/2017/apr/28/federal-police-admit-accessing-journalists-metadata-without-a-warrant

[9] Excerpt from Snowden Archive. Retrieved from https://snowdenarchive.cjfe.org/greenstone/collect/snowden1/index/assoc/HASHc240.dir/doc.pdf

[10] Priest, D. (July 9, 2016). War Reporter Marie Colvin was Tracked, Targeted and Killed by Assad's Forces, Family Says. Retrieved from https://www.washingtonpost.com/world/national-security/war-reporter-marie-colvin-was-tracked-targeted-and-killed-by-assads-forces-family-says/2016/07/09/62968844-453a-11e6-88d0-6adee48be8bc_story.html

[11] Marczak, B. and Scott-Railton, J. (August 24, 2016) The Million Dollar Dissident: NSO Group's iPhone Zero-Days used against a UAE Human Rights Defender. Section 7, Rafael Cabrera. Retrieved from https://citizenlab.org/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/

[12] James, M. et al (January 2008) "Introspection-Based Fault Tolerance for COTS-Based High Capability Computation in Space". 2008 Proceedings of the IWIA. pp 74-83.

[13] Landwehr, A. et al. (2013) "Toward a Self-aware system for exascale architectures." European Conference on Parallel Processing. Springer Berlin Heidelberg.

[14] MIPI Alliance. (August 2012, v2.0) MIPI System Power Management. Retrieved from https://mipi.org/specifications/system-power-management-interface

[15] Huang, A. and Cross, S. (June 13, 2015) Fernvale Main Page. Retrieved from https://www.kosagi.com/w/index.php?title=Fernvale_Main_Page

[16] Michaud, E. and Lackey, R. (December 2013) Thwarting Evil Made Attacks. Retrieved from https://media.ccc.de/v/30C3_-*5600-en-saal_1-201312301245-thwarting_evil_maid_attacks-eric_michaud*-_ryan_lackey

[17] Ross, A. Security Engineering: A Guide to Building Dependable Distribute Systems. "Physical Tamper Resistance". Chapter 14. Retrieved from https://www.cl.cam.ac.uk/~rja14/Papers/SE-14.pdf

[18] https://leaksource.wordpress.com/2013/12/30/nsas-ant-division-catalog-of-exploits-for-nearly-every-major-software-hardware-firmware/

[19] n8fr8. (May 2017, v0.0.8) Phoneypot: Like a Honeypot, but for your phone. Retrieved from https://github.com/guardianproject/phoneypot

[20] Huang, A. (June 2017). Altium Designer 17 source files retrieved from http://bunniefoo.com/bunnie/iengine-tap.zip

[21] King Credie corporate website. Retrieved from http://www.kingcredie.com/. Note: at time of publication, this corporate website is under maintenance.

[22] Huang, A. (June 2017). Altium Designer 17 source files retrieved from http://bunniefoo.com/bunnie/iengine-sam.zip

[23] Wolf, C. and Lasser, M. (2015). Project IceStorm. Retrieved from http://www.clifford.at/icestorm/

[24] bunnie. (March 2017). XZ FPGA test program 1. Retrieved from https://github.com/bunnie/xz-fpga-test1

[25] bunnie, forked from ChibiOS/gdisirio (May 2017). XZ. Retrieved from https://github.com/bunnie/chibios-xz

[26] Amazon.com. (May 2017). Battery Case Ultra Slim 3500mAh Portable Charger Case Extended Backup Smart Phone Charging Case for iPhone 6 6s (White). Retrieved from https://www.amazon.com/Andsun-Mobile-3500mah-Battery-iPhone/dp/B01DGMAGIY, with special thanks to David Cranor and Aqua Jiang for assistance in originally identifying and sourcing this case before it was available on Amazon.

[27] Tor Project. (June 2017) Tor: Bridges. Retrieved from https://www.torproject.org/docs/bridges.

[28] Tor Project. (June 2017) Tor: Pluggable Transports. Retrieved from https://www.torproject.org/docs/pluggable-transports.html.en.