Possessing Mobile Devices

A. A. Adams

Draft: October 23, 2015

Abstract

Smartphones and tablet computers have an ownership model more akin to that of games consoles such as the Sony PlayStation than the PC. Given the ubiquity of these devices and their very broad capabilities and usage, this leaves users vulnerable to significant security and privacy violations. Rather than users possessing these devices, the devices are possessed by multiple third parties, to the detriment of users' rights.

1 Introduction

Although modern smartphones and tablet computers are at least as powerful as the PCs of a decade or so ago, they are conceived of as primarily media consumption and communication devices. As such, and despite the fact that their associated hardware such as built-in cameras, microphones, accelerometers, GPS and other location sensors etc. all pose significant privacy risks, the ownership model for these devices owes more to that of the home gaming console than that of the PC. According to various sources (IDC [12] and Strategy Analytics [18]) more than 1 billion smartphones running the Android operating system were shipped in 2014 (shipped means sent out from manufacturers not necessarily actually sold to or used by consumers). This billion devices accounted for over 80% of the market in 2014, and combined with over 192 million iPhones shipped (approx 15% of the market) Android and iOS phones account for 96% of the smartphones shipped in 2014. These figures account for only smartphone shipments, not tablets. In addition, reports from 2014 also concluded that the majority of online information exchange in the US now takes place via mobile devices [20, 16]. Users pay significant amounts of money for these devices and conduct much of their social life and business using them. But despite having paid for them, owners' rights over these devices are limited, unless they bypass the built-in software restrictions and *root* their Android device or *jailbreak* their iOS device. So, it appears that despite having paid for them, they may not truly *own* these intimate devices.

In this article I explore these ownership issues, where they originate, and their implications for the security, privacy and autonomy of users and the economic and ecological implications. In particular, I argue that the current model of smartphone/tablet ownership violates reasonable expectations and fundamental rights of users without giving them sufficient recompense. Although the device is bought and, therefore, supposedly owned by a user, the manufacturer/software system integrator/retailer/network connection provider retains much of the control of the device and both prohibits the user from protecting their privacy and from making use of the device's full capabilities. I also argue that the claim that such external control is improving the security of users is false in multiple ways, in particular due to the slow delivery of patched versions of the operating system, and the speed with support for offering patches at all is dropped for specific models.

Ownership is not as simple a concept as it may first appear. There are legal concepts of ownership which confer both rights and responsibilities on the owner(s). There are psychological elements where individuals may feel their rights have or even their person has been

violated where their legal rights or technical control over things they own do not match their expectations. There are economic issues driving the market to restrict the abilities the owner of a device may exert. In the end what we think of as ownership is simply a shorthand for a bundle of rights in an object. Ownership rights usually include the right to decide who can use an object. Something which is owned can usually be sold to another. However, in most countries one cannot sell body organs, even those without which the donor can survive (one kidney or part of a liver) despite most people considering that they own their body [9]. Such restrictions are often justified by appeal to a general social benefit, such as avoiding exploitation of the poor as a resource of body parts by the rich. In the case of smartphones and tablets, however, these ownership restrictions seem far from justified when one considers the cost to users in terms of both privacy and security.

2 Psychological Attachment to Personal Devices

Smartphones are phones and also computers. To understand user expectations of ownership, therefore, we need to consider the background of ownership rights in both. Since this discussion is of personal devices, I focus on the PC era for computers and (mostly) the mobile telephone era for telephones. One of the differences that personal computers brought was not just the ability for people to have computers in their homes, but the installation of a device in their individual working space which even though owned by the organisation was described to be, and felt to be, a *personal* device. As Reeves and Nass [21] describe, people's emotional/psychological attachments to devices are often quite illogical, such as making distinctions between completely fungible devices (identical specification, all data stored on a network) based simply on prior usage of that particular machine.

Early telephone networks in many countries only allowed devices supplied by the network operator to be connected to the network. They claimed this was to prevent damage to the network, although as the US Carterfone [11, 28] case demonstrated, this was at least partly a spurious claim and in fact the preservation of sales or rental income on monopoly-provided equipment was the primary reason. Pre-smartphone mobile phones in the developed world quickly became objects of deep emotional attachment for their owners [25].

Given the intense and intimate usage of modern mobile devices, we would expect that users will develop strong positive feelings including trust, towards their devices. As we shall see, however, this trust is misplaced because of the lack of controls on the real "owner" of the devices: the provider(s) (manufacturers and mobile telephone operating companies primarily) who retain a great deal of control.

3 Technical Ownership (Control) of Mobile Phones

Early digital mobile phones had very limited capabilities beyond making phone calls and sending/receiving short text messages. As their capabilities expanded to use of digital cameras and connect to networked information services, the hardware and operating systems became more complicated and issues of interoperability between networks and phones, and between phones and other devices (particularly PCs) came up. Early featurephones featuring information services ran a variety of operating systems, with different levels of openness. Most early featurephones included limited or no ability to update the system software, and in particular "Over the air" (OTA) updates to the phone's firmware (the downloading of an updated core operating system via the mobile network) were not generally supported. Many phones ran highly customised operating systems and there were few systems used by more than one manufacturer.

Firmware-installed operating system upgrades, if possible at all, were generally restricted to special purpose hardware at service centres. Some later phones allowed the user to update

by downloading new firmware to a PC over the Internet, connecting the phone to the PC and running an update program on the PC to re-write the phone's software. (This was also the early form of updates for Apple's iPhones until iOS 5 which introduced the OTA update process.)

The road from the digital mobile handset to the smartphone had many dead ends, byways and failed highway projects. The smartphone basically combines a digital mobile phone handset and a personal digital assistant (PDA). The openness or closure of many of these early attempts reflects whether the creators started from a phone and tried to give it PDA functionality and Internet access, or from a PDA and tried to give it phone functionality and Internet access. So, for example Nokia mostly started from a PDA concept and created environments such as the S60 platform and the Symbian system (which superseded S60 at Nokia and which was based on the EPOC operating system from the UK's Psion PDA maker) while Microsoft developed the Windows CE/Windows Mobile system.

In Japan the former state fixed line telephone monopoly provider NTT developed MOAP (Mobile-Oriented Applications Platform) systems — one based around a Symbian kernel and the other around a Linux kernel — did not have open third party development options, nor user-installable applications. They used NTT's proprietary *i-mode* system to provide Internet-like services including translation of suitable web pages to a form viewable on the greyscale phone screen and to use the keypad for interaction. As with computer gaming consoles, these systems did have application development platforms, but access to these required development companies to enter contracts with NTT. Application development for these systems was typically done by or under contract to the hardware manufacturer, seeking to compete in the market by offering built-in applications. Japanese rivals "au by KDDI" and "SoftBank Mobile" developed phones supporting the WAP (Wireless Application Protocol) standard which allowed access to websites through a style sheet-like approach. Interactive applications running locally, however, could still only be produced using proprietary software development kits. Email on ll of these Japanese phones, for example, was only available through dedicated apps using the service provider's mail server, or through a WAP-enabled web mail service.

Systems with open application development environments such as PalmOS, its successor WebOS, Symbian, Blackberry OS, iOS and Android have gradually taken much of the market share for mobile devices including not just smartphones but the larger tablet computers. Devices running these systems are, as I have said, really general purpose computing devices, with mobile networking and integration with the standard POTS (plain old telephone service) via a "phone" app. They are generally designed to be devices with which software is used, rather than on which software is developed and although there are some applications (such as TerminalIDE for Android) in which programs can be developed, these devices are not intended as platforms on which to develop apps to run on them. Most development happens on other more powerful computers running suitable development tools.

There has been and remains a variety of levels of openness in these systems with regards to user control. iOS devices are generally designed to only allow applications to be installed from the Apple App store. Android has an option for vendors to preset certain application sources as allowed and disallow others. In some distributed versions other sources may be switched on by user control, but some distributed versions limit the application sources to only the preset ones. Versions of RIM's BlackberryOS before 10 (which was a complete rewrite based on the QNX kernel) restricted application installation to only RIM's repository. The Blackberry 10 system, however, features support for Android applications including the ability to install such applications from alternative sources like the Amazon Appstore for Android. Windows Phone devices are limited to installing apps from the Windows App store.

Anyone who has physical access to a device can, with enough effort, typically control that device. Physical access restrictions are a standard part of security engineering [1, Ch. 11]. However, most ordinary people do not have the expertise and/or equipment to work around

built-in control restrictions on devices. There are also sometimes legal restrictions on doing so which make it illegal to do so [15], more difficult to obtain the required hardware [6], or place the user in breach of a contract to do so [27].

While some manufacturers such as Sony and Asus provide instructions and options for users to access full administrative rights (root user or superuser) on some of their Android-based devices, they do so only with the agreement of the mobile network provider in many cases, which is often withheld. Other manufacturers and many mobile network operators pre-load Android devices with (often unwanted) apps which are not deletable on a non-rooted phone (referred to as bloatware). Many of these apps are also set to start on boot, requiring users to remember to manually turn them off after every re-boot — the option to not run on boot is also usually locked out from user settings.

A recent interesting development regarding bloatware on mobile phones was the launching of a lawsuit in China by a small consumer protection group (the Shanghai Consumer Council) which sued Samsung and Chinese vendor Oppo for violating consumer rights by selling them devices with undeletable bloatware installed [13].

Regulators such as the FCC in the US have been reluctant to require manufacturers and network operators to grant users with full control over their own devices due to concerns about the potential for misuse of software defined radio capabilities to interfere with other devices, both other mobile phones and other radio communications. However, neither the US Copyright Office's exemption of jailbreaking iPhones and rooting Android phones [15] from violating the DMCA's anti-circumvention rules, nor the prevalence of these practices by users have persuaded telecoms regulators to insist that users be given real ownership and control over their devices.

4 Security and Privacy on Possessed Devices

The primary uses of smartphone and tablets are for communication (SNS, photo sharing, messaging, voice/video calls), although media consumption (games, video, audio, text) and information processing (note-taking, self-quantification) are also significant uses. The locked-down model of the previous generations of media consumption devices, whereby the manufacturer, or other upstream retailer, has significant control of the device, seems a poor deal for consumers. Schneier called this the "feudal security" model in blog posts (`http://tinyurl.com/b7s2fq4` and `http://tinyurl.com/k8x5de4`). As in the feudal social model, the overlords are not trustworthy and the moral hazards of their position without strong external regulation leads them to abusive practices such as spying on users' locations without their knowledge (Google: [4], Apple: [14]). Device manufacturers, meanwhile are constantly tweaking proprietary device drivers for their Android phones [29], shipping binary blobs for attachment to Android's Free Software Linux kernel, with too little appreciation of the security risks of these often hastily programmed hardware interfaces.

4.1 Direct Security Risks of Rooting/Jailbreaking

Liebergeld and Lange [17] discuss the risks that users run if they root their Android devices while Rogers [22] provides a similar discussion of some of the dangers of jailbreaking an iOS device. Since neither Android nor iOS are designed to have administration accounts running, despite being both based on Unix-related kernels (Linux and XNU, respectively) once the systems have been hacked to expose these administrator-level accounts, they are more likely to be vulnerable to external hacking. While users' privacy and to some extent their security are always at risk from any application they install (and from other vectors), once they have rooted/jailbroken their device, applications they install can now request root access and many are likely to grant it just as they grant privacy-invasive privileges to apps such as those to control the camera flash as a flashlight [23].

4.2 Indirect Security Risks of Rooting/Jailbreaking

The hoarding of vulnerabilities by the NSA and GCHQ (and probably many other SIGINT agencies) has been condemned by security professionals as putting the security of everyone at risk from criminals by decreasing the chance of the project management becoming aware of the vulnerability and taking steps to fix it [3]. Similarly, since jailbreaking an iOS device or rooting many Android devices requires breaking their security model, users (particularly highly skilled white hat hackers) have an incentive to prevent the system developers from knowing about the vulnerabilities they exploit. These vulnerabilities, in addition to being used by users to gain control over their machine, can also potentially be used by attackers to elevate their privileges as part of a malicious attack.

In addition, by preventing users from controlling their own devices, users are encouraged to try to follow instructions on how to bypass the security on their device from dubious sources. While most of the directions online about jailbreaking/rooting device are just what they appear to be, most users do not have the technical expertise to know when what they are doing will actually achieve their goals or not, and whether in doing so they are instead (or in addition) installing some form of malware or opening up a security hole in their device. Such attacks are known to be targetted at Facebook users; Facebook have named this the self-xss attack (self-cross-site-scripting).[1] This willingness of users to follow somewhat random online advice on how to break the security on their devices is not something which should be encouraged, any more than car owners should be encouraged to install updates to their car's onboard systems using a USB stick delivered to their address without verifying its source as the manufacturer [7].

4.3 Security Risks of Not Rooting/Jailbreaking

Without administrator control of a device, checking the integrity of system files and monitoring the presence and activity of installed applications is very difficult. On both iOS and Android, in fact, ordinary user-space applications are not supposed to monitor or interfere with other apps. Google and Apple enforce such policies in their respective application stores, although for most Android devices one can install apps from other sources. Even where such monitoring can be installed as a user-space app its access to the activities of other software is limited.

The lack of administration control becomes more of a problem over time as smartphone providers (manufacturers, system integrators, telcos etc.) all seem to want to push users into upgrading their devices more often than some would wish to do. With the rapid pace of development of iOS and Android and the rapid development of new models, there is a problem with older devices not being provided with updates by the system providers (typically the manufacturers). Even where devices are still supported by manufacturers, these updates are being rolled out far too infrequently. A recent study by Thomas, Beresford, and Rice [24] showed that, even where Google is patching the base Android system, many manufacturers are very slow at feeding such patches through to users' devices, with 87% of Android machines in their study having known unpatched vulnerabilities.

Once updates for the core iOS or Android system stop appearing, devices often cannot run updated versions of various applications, leaving them vulnerable to security problems in the older versions of apps as well as in the operating system itself. A very serious version of this problem appeared in January 2015 when Google announced that it would not itself provide a security fix for a known vulnerability in the WebKit web browser app which was a key element of Android 4.1 to 4.3 [2] (although Google did say they would accept and push a patch if offered by a reliable third party). While it is possible to use alternative browsers such as Mozilla's Firefox, which is updated and available even on these older Android versions, many other apps use the WebKit rendering engine for their own

[1] www.facebook.com/notes/facebook-security/dont-be-a-self-xss-victim/10152054702905766

html parsing and presentation. As noted above, users find it difficult or impossible to know which apps interoperate with which other elements of the system, particularly core elements such as the web rendering engine.

Unlike, for example, PCs running Windows XP which was supported by Microsoft with security patches for over a decade, Android 4.3.1 was only released in October 2013. Users are generally completely at the mercy of the hardware manufacturer to compile and release a new version of Android for their hardware and so it is likely that phones released in mid-2013 might not have had an upgrade offered by the manufacturer beyond 4.3.1, which less than eighteen months later had security vulnerabilities in a core service app which Google decided not to patch, and which even if patched by Google would probably not be offered as a downstream update by other manufacturers.

Without gaining administrative access, which smartphone providers are reluctant to grant to users, Android users cannot install even an alternative compatible operating system such as CyanogenMod. iOS device users are faced with similar situations with their reasonably recent devices (sometimes less than two years old) being left out of the operating system upgrade cycle and therefore forced to upgrade their hardware or remain vulnerable. Even for a jailbroken iPhone there appears to be no alternative operating system that can be installed to make up for the lack of an Apple-provided security-updated iOS.

4.4 Privacy Risks of Not Rooting/Jailbreaking

Security and privacy are often represented as oppositional duals: one must give up some privacy in order to gain some security. While this may be true in some circumstances, the security of the devices that one uses is a pre-requisite for privacy, not in opposition to it. Just as with the lack of ability to see whether unauthorised software is running requires administrator access, so does monitoring or controlling the provision of private information by applications. Android applications such as Android Privacy Guard (APG) require root access to provide such facilities to users.

5 So Who *Does* Own My Device?

So, ownership is not a single absolute concept granting all possible rights over an item. However, the locking down of smartphones, whose hardware such as microphones, cameras, accelerometers, GPS and whose software and data such as contact listings, photos, social network posts, email, communications and media consumption, make them so useful but also so risky in terms of both privacy and security, are not primarily owned by their users. Instead, the telephone company, hardware manufacturer and system integrator are the practical owners of these devices.

In recognition of that, then at the very least, such lack of ownership requires a significant improvement in consumer rights and privacy protections. As the work of Thomas, Beresford, and Rice [24] shows, manufacturers of Android smartphones are leaving their users with vulnerable software due to unwillingness to provide regular updates. In the world of PCs, patching has become one of the standard backbones of ensuring security. A system administrator who does not patch their systems is regarded as unprofessional at best and even as criminally negligent. Home users are exhorted to keep their systems up to date and Windows 10 Home Edition no longer allows users to defer security updates in an effort to preserve the security of the ecosystem.

There is, however, a long history of software being provided "without warranty". Consumer goods such as cars and drinks used to be similarly outside such claims of negligence in most circumstances, but seminal court cases in the early twentieth century established a duty of care for manufacturers to not sell dangerous goods into the supply chain, such as

cars with faulty brakes [26] (US: MacPherson v. Buick Motor Co.) or drinks contaminated with slugs [10] (Scotland, UK: Donoghue v. Stevenson). The implications of MacPherson v. Buick are likely to to become even more important as cars become further informatised and, even without being driverless, become more and more vulnerable to external hacking [8]. While not usually quite so physically dangerous, smartphones and tablets are now so embedded in our lives that their information is a vital part of our personal infrastructure and the lack of liability of manufacturers/telcos/retailers is becoming hard to defend.

At best the US Copyright Office's exemption of iOS jailbreaking/Android rooting from illegalisation under the DMCA [22] should be extended in the US and adopted elsewhere as a clear right of device owners to simply and easily opt out of external controls by others (whether that be a person or an organisation) on any device and an insistence that the owner of the hardware should have at least full visibility of the operation of their device and really a much greater level of control, that is to say, proper ownership of the device. Remaining limitations should be clearly justified in the public interest, and not simply in the commercial interest of providers (reducing their costs by not bothering to issue security updates, allowing them to charge users for permissions to make use of innate capabilities of the device, or profit from the invasion of users' privacy). Where support for security updates on a device are no longer offered, then no restrictions on user access to full control of the device is justified. At that point, perhaps, legal liability for failures might shift from providers to users, much as it already does with PCs. Those still running Windows XP have only themselves to blame if their devices invade their privacy or are used as zombies in a botnet.

Unfortunately, the direction of travel does not seem to be in this direction. The US Copyright Office demurred from extending their "right to jailbreak" from iPhones and Android phones to iPads and Android tablets in 2012 [5]. The latest leaks of the controversial and secretly negotiated (and supposedly to be kept secret from the public until after legislatures have voted on whether to accept it) Trans-Pacific Partnership (TPP) include requirements on further locking down devices and harsh penalties for anyone found circumventing technical protection measure, including destruction of the machine. If this agreement is adopted, then in places like the US, Australia and Japan, a rooted Android or jailbroken iOS device which could bypass the DRM on music, books or video files [19].

References

[1] R. Anderson. *Security engineering*. 2nd ed. John Wiley & Sons, 2008.

[2] P. Bright. *Google won't fix bug hitting 60 percent of Android phones*. 13th Jan. 2015. URL: tinyurl.com/o2d5hho.

[3] M. D. Cavelty. "Breaking the cyber-security dilemma: Aligning security needs and removing vulnerabilities". In: *Science and engineering ethics* 20.3 (2014), pp. 701–715.

[4] R. Chow. "Why-Spy: An Analysis of Privacy and Geolocation in the Wake of the 2010 Google Wi-Spy Controversy". In: *Rutgers Computers and Technology Law Journal* 39 (2013), pp. 56–93.

[5] Copyright Office of the Library of Congress (US). *Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies*. 2012. URL: tinyurl.com/ngewrn2.

[6] B. F. Fitzgerald. "The PlayStation mod chip: a technological guarantee of the digital consumer's liberty or copyright menace/circumvention device?" In: *Media and arts law review* 10.1 (2005), pp. 85–98.

[7] N. Ford. *Hacked Jeep USB software patch criticized*. 8th September. 2015. URL: tinyurl.com/qy82237.

[8] A. Greenberg. *Hackers Remotely Kill a Jeep on the HighwayWith Me in It*. Wired, 21st July. 2015. URL: tinyurl.com/oaabx46.

[9] J. W. Harris. “Who owns my body”. In: *Oxford Journal of Legal Studies* 16.1 (1996), pp. 55–84.

[10] V. Heuston R. F. “Donoghue v. Stevenson in retrospect”. In: *Modern Law Review* 20 (1957), p. 1.

[11] M. T. Hoeker. “From Carterfone to the iPhone: Consumer Choice in the Wireless Telecommunications Marketplace”. In: *CommLaw Conspectus* 17 (2008), p. 187.

[12] IDC. *Android and iOS Squeeze the Competition, Swelling to 96.3% of the Smartphone Operating System Market for Both 4Q14 and CY14, According to IDC*. Press Release from International Data Corporation. 2015. URL: tinyurl.com/p5mltv4.

[13] M. Kan. *Samsung faces lawsuit in China over bloatware on phones*. PCWorld 3rd July. 2015. URL: tinyurl.com/pqqddke.

[14] V. Kumpu. “Privacy and the emergence of the “ubiquitous computing society”: The struggle over the meaning of “privacy” in the case of the Apple location tracking scandal”. In: *Technology in Society* 34.4 (2012), pp. 303–310. ISSN: 0160-791X. DOI: dx.doi.org/10.1016/j.techsoc.2012.10.002. URL: www.sciencedirect.com/science/article/pii/S0160791X12000565.

[15] T. B. Lee. “Jailbreaking now legal under DMCA for smartphones, but not tablets”. In: *arstechnica* (2012). 26th Octover. tinyurl.com/8os3qn5.

[16] A. Lella and A. Lipsman. *The U.S. Mobile App Report*. comScore White Paper. 21st August. 2014. URL: tinyurl.com/pokl2uf.

[17] S. Liebergeld and M. Lange. “Android security, pitfalls and lessons learned”. In: *Information Sciences and Systems 2013*. Springer, 2013, pp. 409–417.

[18] N. Mawston. *Android Shipped 1 Billion Smartphones Worldwide in 2014*. Strategy Analytics Report. 29th January. 2015. URL: tinyurl.com/om9etpe.

[19] J. Pearson. *White Hat Hackers Would Have Their Devices Destroyed Under the TPP*. Motherboard, 9th October. 2015. URL: tinyurl.com/o3bm553.

[20] S. Perez. *Majority Of Digital Media Consumption Now Takes Place In Mobile Apps*. TechCrunch. 21st August. 2014. URL: tinyurl.com/mlvo5el.

[21] B. Reeves and C. Nass. *The Media Equation*. 2nd Ed. Stanford: CSLI, 2002.

[22] K. Rogers. *Jailbroken: Examining the policy and legal implications of iPhone jailbreaking*. 2012.

[23] SnoopWall. *Flashlight Apps Threat Assessment Report*. 2014. URL: tinyurl.com/pvj3oh3.

[24] D. R. Thomas, A. R. Beresford, and A. Rice. “Security Metrics for the Android Ecosystem”. In: *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM. 2015, pp. 87–98.

[25] J. Vincent. “Emotional Attachment to Mobile Phones: An Extraordinary Relationship”. In: *Mobile World*. Ed. by Lynne Hamill, Amparo Lasen, and Dan Diaper. Springer London, 2005, pp. 93–104. URL: dx.doi.org/10.1007/1-84628-204-7_6.

[26] J. W. Wade. “Strict Tort Liability of Manufacturers”. In: *Southwestern Law Journal* 19 (1965), p. 5.

[27] M. H. Wolk. “The iPhone Jailbreaking Exemption and the Issue of Openness”. In: *Cornell Journal of Law and Public Policy* 19 (2009), pp. 795–828.

[28] T. Wu. “Wireless Carterfone”. In: *International Journal of Communication* 1 (2007), pp. 389–426.

[29] X. Zhou et al. “The peril of fragmentation: Security hazards in android device driver customizations”. In: *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE. 2014, pp. 409–423.