When Mobile is Harder Than Fixed (and Vice Versa): Demystifying Security Challenges in Mobile Environments

Jon Oberheide
Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
jonojono@umich.edu

Farnam Jahanian
Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
farnam@umich.edu

ABSTRACT

Sophisticated consumer mobile devices continue to approach the capabilities and extensibility of traditional computing environments. Unfortunately, these new capabilities and applications make mobile devices an enticing target for attackers and malicious software. Due to such threats, the domain of mobile security has been getting a considerable amount of attention. However, current approaches have failed to consider key differences and their practical impact on the security of modern platforms when adopting techniques from non-mobile (or "fixed") environments. To help demystify mobile security and guide future research, we examine the unique challenges of mobile environments ranging from hardware to software to usability, delve in the diverse security models of current mobile platforms, and present our five commandments of mobile security research.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection

General Terms

Security, Reliability, Management, Performance

Keywords

Mobile Devices, Malicious Software, Security Models

1. INTRODUCTION

Modern mobile platforms are reinventing the mobile landscape. These devices run commodity operating systems and have complete multi-protocol networking stacks, UI toolkits, file systems, and other fully-featured libraries. While past mobile platforms had limited functionality and were relatively closed to third-party applications and user extensibility, new mobile platforms ship with complex Internet, productivity, communication, and application suites and strongly encourage third-party development with comprehensive software development kits and application delivery mechanisms.

The Eleventh International Workshop on Mobile Computing Systems and Applications February 22, 2010, Annapolis, MD, USA

However, not all is roses, as these mobile devices face a wide range of new security challenges and malicious threats. The same extensibility that has enabled rich functionality and applications has also made them an enticing target for attackers. These devices are increasingly used to store sensitive personal information such as financial data used for mobile banking, not to mention the potential abuse for snooping on a mobile user's voice, SMS, data, and location services.

In recent years, researchers have recognized the importance of mobile security [16, 5, 3]. Such research is often polarized between one extreme of blindly taking existing security approaches and applying them to mobile platforms without considering their subtle differences and the other extreme of claiming that fundamentally new techniques are necessary to protect mobile devices. The truth, we believe, lies in between. In addition, previous work that has explored security-related differences between fixed and mobile environments will often enumerate a list of mobile attributes without going in-depth on the resulting impact on practical security considerations and the diverse range of security models of current platforms.

In this position paper, we aim to demystify some of the common misconceptions in the mobile security domain and shed light on the challenges that lie ahead. As was the case with virtualization security [6], we believe there is a need to explore the domain of mobile security in a comprehensive manner as the security concerns with mobile platforms continue to increase.

However, before we dive in, it's important to clarify the scope of our position paper as the term "mobile security" can refer to a broad range of topics. Our research focuses on the security threats faced by mobile consumer electronic devices (CEDs); in particular, the potential for widespread malware infection and malicious applications, similar to those security problems that continue to plague our fixed computing platforms. While much research has looked at using mobile devices as a trusted device or authenticator [13], we only concern ourselves with security of the devices themselves rather than security applications that the devices may enable. While we do not claim to be exhaustive in our enumeration of potential challenges or threats, we place particular focus on the topics that we see as most critical in the future and believe to offer new insights beyond previous work.

The rest of our paper is structure as follows: in Section 2, we examine the unique challenges of mobile environments and discuss what practical impact they have on the security of mobile devices; in Section 3, we classify and discuss the diverse security models of current popular mobile plat-

forms; and in Section 4, we present our five commandments of mobile security research as light-hearted "suggestions" for avoiding common pitfalls in future research endeavours.

2. SECURITY CHALLENGES IN MOBILE ENVIRONMENTS

Mobile environments have unique differences from traditional fixed computing environments. While some differences are straightforward, others may have subtle consequences that can have a significant impact on the security of a mobile device. Rather than simply compiling a laundry list of how mobile environments differ, we focus our discourse around the resulting security impact and discuss the current and future challenges and opportunities that arise.

2.1 Resource Constraints

The most obvious challenge for developing security mechanisms for mobile devices is their constrained resources. Unlike their desktop brethren, mobile devices have strict resource constraints in both computational and power capabilities due to their mobility and small size. Therefore, while complex detection algorithms may scale in standard non-constrained desktop environments, they can be less effective in resource-constrained mobile environments.

Malware Detection While security vendors have marketed mobile-specific versions of antivirus software to detect malware, these solutions are similar to their desktop variants and provide limited detection capability with significant power and resource overhead. Even with simple signature-based static analysis, the computational resources required to perform such analysis can be high. For example, the ClamAV antivirus engine available for the Nokia N800 mobile device requires 57 seconds of processing just to initialize its signature database and consumes as much as 40 megabytes of memory [10]. In addition, more computationally expensive algorithms such as behavioral detection engines, which are becoming more important for detecting sophisticated threats, are simply infeasible to deploy on current mobile devices due to their heavyweight resource requirements. Therefore, adapting traditional approaches to malware detection may not be appropriate for the mobile environment as they consume a significant amount of on-device resources and power.

Scalability While the number of threats currently targeting mobile devices is significantly lower than the threats faced by traditional desktops, this number will certainly grow as the use of modern mobile devices becomes ubiquitous. If mobile attacks follow a trend similar to traditional fixed computing, they will may face a deluge of new threats, requiring detection systems that can scale elegantly to handle a diverse and sophisticated threat landscape. Existing research has proposed moving the complexity of mobile malware detection off-device to a network service to allow transparent extensibility and scalability [10, 9].

2.2 Mobile Attacks

Mobile Botnets Compared to traditional fixed computing, the case for mass ownage of mobile devices for creating a botnet is not as straightforward. Traditionally, attackers are able to monetize their botnets of compromised hosts through spam, denial of service extortion, sensitive data theft, and phishing. Compromised hosts are often considered valuable to an attacker if they have high-throughput, low-latency, and stable connectivity to the Internet and have significant system resources, attributes that are not common with today's mobile devices. However, as more and more sensitive data such as login credentials are stored on mobile devices, attackers may still wish to target them for harvesting data. Given the day-to-day reliance on mobile devices that people have for personal communications, ransomware attacks may also be particularly effective and have already targeted users en masse in China.

Attacker Incentives It's unclear at what point it will become worthwhile for attackers to expend serious effort to target mobile platforms. While mobile platforms offer a new, unexplored attack surface with local and wide area radios speaking previously unfuzzed protocols, attackers are having their way with traditional platforms and may not be sufficiently incentivized to change up their ways. Measuring the incentive for an attacker to target a particular platform is quite a difficult task. Existing work has employed game theory models based on market share to determine at what point attackers will expend significant effort to target OS X compared to Microsoft Windows [11]. It's entirely possible that we'll see a repeat of the early 2000s when we saw flash worms that weren't overtly malicious in their activity but rather were simple proof-of-concepts. It wasn't until underground markets developed and attackers realized the potential for economic gains that more serious and sophisticated malware was developed. It's not unlikely that'll we'll see a similar series of events as attackers "feel out" the mobile space and pinpoint the approaches that will result in maximal monetary gains for the effort expended.

Targeted Attacks Rather than mass ownage for the purpose of building botnets, mobile devices may represent a more enticing mark for targeted attacks. Compromising a particular individual's mobile device is a one-stop-shop for attackers to snoop on their voice/SMS/data communications, track their physical location in real-time via GPS functionality, and even eavesdrop on non-cellular conversations via the device's microphone. Compromised mobile devices may also act as bridges to previously unreachable locations such as when a worker brings his device into work or plugs it into his PC, allowing penetration of an enterprise's network perimeter. As is true in the world of traditional computing, targeted attacks are not only more severe in their impact but are significantly more difficult to detect and mitigate than your garden-variety malware aimed at mass ownage.

2.3 Architectural Considerations

Attributes of a platform's hardware architecture can have an impact on the overall device security. While certain constraints of hardware architectures may make mobile security more challenging, there can be positive security-enhancing attributes as well. For example, the ARMv6 core offers security features such as TrustZone which can be employed to separate software into trusted and untrusted virtual worlds and XN (eXecute-Never) support to mark non-executable memory pages. Unfortunately, many mobile platforms do not take advantage of these security features provided by the architecture.

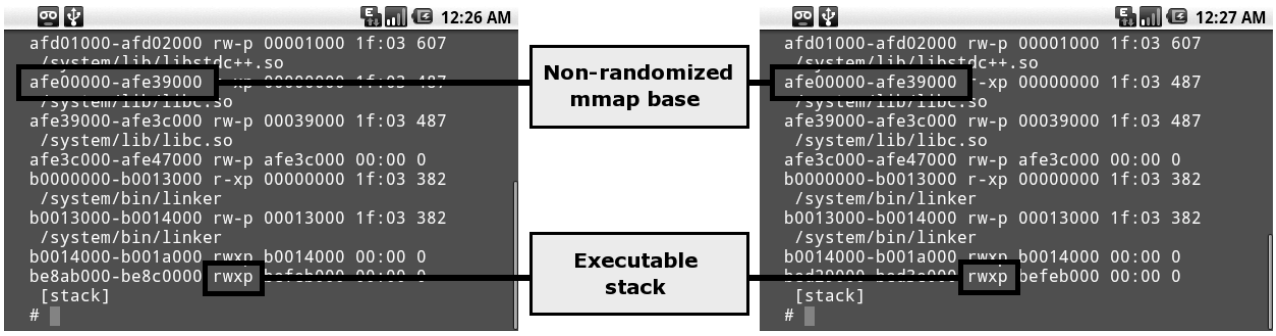

Figure 1: A lack of effective ASLR and NX support, as seen here in `/proc/$pid/maps` output of Android processes, is unfortunately a common occurrence on many mobile platforms.

ASLR Address space layout randomization (ASLR), an effective technique that thwarts common memory corruption attacks by randomizing the location of the stack, heap, mmap base, and other elements within a process address space, is often weak when applied to mobile devices due to architectural concerns. As current mobile devices do not need to address a large amount of physical memory, they commonly have 32-bit CPUs. As the effectiveness of ASLR hinges solely on having an adequate amount of randomized bits in memory addresses, a 32-bit address space can be ineffective at preventing attacks in many scenarios. The problem is compounded as mobile platforms often respawn crashed applications in order to maintain a seamless user experience, giving attackers the ideal scenario for trivial bruteforcing of a 32-bit ASLR implementation.

Trusted Computing Some hardware features have potential for a positive impact on mobile security. For example, trusted computing capabilities, such as the Mobile Trusted Module [4], may assist service providers or other parties in verifying the integrity of mobile devices for regulatory or security applications. Some of the issues that hinder widespread adoption of trusted computing functionality by traditional computing, such as public key infrastructure (PKI) scale and management, may be less of a challenge in mobile environments as the trusted computing base may be smaller and less volatile in nature [15].

Virtualization Lastly, hardware support for virtualization offers exciting opportunities for mobile security. Mobile hypervisors [2] may be used to separate virtual domains to ensure isolation at an architectural level. Many corporate users usually either carry around a "work" phone and a "personal" phone or risk using a single device. With mobile virtualization, a work domain can be isolated from a personal domain, allowing the user to install whatever applications and games they desire without putting their corporate data at risk.

2.4 Platform and Network Obscurity

A lack of visibility into mobile platforms and networks can pose a challenge for security researchers. Despite current conditions, we expect to see progress in the area of mobile platform and network obscurity as consumer, developer, and researcher demand increases openness.

Platform Obscurity While many mobile platforms are based on commodity operating systems (e.g., Linux-based Android/Moblin/webOS, OS X-based iPhone), they can look significantly different from a security perspective and require novel analysis. In addition, platforms are often intentionally restricted from modification and instrumentation due to mobile carrier agreements and regulatory requirements.

Auditing, Debugging, and Forensics Regardless of whether a platform is intentionally closed or just simply has not been analyzed in-depth due to its recency, obscurity can hamper the efforts of both attackers and defenders. While emerging platforms offer a new attack surface ripe for exploitation by malicious parties, vulnerability discovery tools and exploitation techniques may need to be adapted [12]. As always, the challenge of analyzing and attacking a new platform brings out some of the best talent in the research community. For example, researchers developed SMS fuzzing frameworks for several mobile platforms to locally fuzz SMS parsing routines instead of interacting with the device via its radio [8]. The fuzzing uncovered remotely exploitable vulnerabilities in the iPhone's SMS routines. Defenders also must adapt their traditional software toolsets for debugging, auditing, reverse engineering, and forensics to mobile platforms. Given the complexity of such tools and variety of platforms and security models, this is a non-trivial task.

Endpoint Management The lack of visibility into a mobile platform can also hinder fine-grained management capabilities, often reducing overall security. This is especially true in enterprises where endpoint management is vital and every device attached to the network must be identified, audited, and maintained. Untrusted mobile devices with potentially compromised applications may provide attackers with effective avenues for infiltrating enterprise networks. Enterprise-focused platforms, such as RIM's Blackberry, have started to provide some management capabilities on their devices.

Network/Protocol Obscurity Telecommunication networks are infamous for their penchant for "security by obscurity" and mobile networks are no different. The internals of mobile network operation and protocols tend to be kept closely by service providers and not publicly available to researchers for analysis and audit. As a mobile network operator may maintain control capabilities over the mobile handsets on its network, the loss of integrity of various compo-

Mobile Platform	Application Delivery	Trust Levels	System Isolation
Apple iPhone	high	low	low
Google Android	medium	high	high
RIM Blackberry	low	medium	low
Symbian OS	medium	high	medium
Windows Mobile	medium	medium	medium

Table 1: A ranking of the security attributes of the 5 most popular mobile platforms. The high, medium, and low scores refer the level of protection provided for each category (e.g., "high" is a positive ranking).

nents of the mobile network may directly affect the integrity of the end user devices and applications. In addition, a lack of visibility in mobile network operations can make the development of defensive measures a challenging task without the cooperation of a mobile provider.

We continue to see researchers chiseling away at these walls of obscurity and having considerable success. For example, numerous researchers [7] have detailed practical attacks against the A5/1 cipher, allowing attackers to intercept and decrypt GSM communications. Projects such as OpenBSC can provide researchers with a stable network platform to perform such experimentation without disrupting provider networks or violating legal regulations. Also, in [14], researchers demonstrated effective denial of service attacks against mobile network HLRs. The ease at which such attacks can be performed once the veil of obscurity is lifted is certainly cause for concern.

2.5 Mobile HCI/Usability

Mobile environments tend to have very different application installation and usability patterns from desktops and workstations. On traditional desktop systems, many of the common tasks a user performs are available through rich web services. For example, a user can check their email, read the news, do their banking, and construct presentations and spreadsheets all through their web browsing application. Since desktop systems typically have traditional keyboard and mouse interfaces with large screens, interacting with these common applications in a web browser environment is not a difficult task.

Application Usability Mobile devices are often limited by their input and display capabilities. Even advanced devices that have QWERTY keyboards and touch screens do not approach the input ease of a full keyboard, mouse, and display. As a result, using the same rich web services such as email, banking, and office apps within a mobile web browser is extremely difficult. To address this problem, developers often have to create custom applications that use the native UI capabilities of the device to make interacting with a web service more usable. For example, there are custom applications available in the Android Marketplace for numerous common websites like Amazon, Facebook, Gmail, The Weather Channel, and Wikipedia.

Privilege Separation Prompted by the different usability patterns in mobile environments, the use of numerous individual applications on a mobile platform can be a double-edged sword when it comes to overall security impact. On the positive side, a platform that provides sufficient isolation mechanisms between applications can benefit from having independent applications for each user task. By following the principals of privilege separation, a platform can reduce the impact of a buggy or compromised application by separating functionality into multiple individual applications. For example, using a single vulnerable web browser application to access a banking website and a potentially malicious website may result in an attacker gaining access to the user's banking credentials. However, if the mobile platform provides isolation and the banking functionality is offered as a standalone application, the compromise of the web browser may not affect the user's banking credentials. This separation of privilege has been intentionally employed in scenarios such as the Google Android media player, which separates out notoriously vulnerable multimedia codecs from the rest of the media player functionality [1].

User Conditioning The abundance of these individual applications may have a detrimental effect on the security of mobile platforms that do not offer strong isolation. Users may become conditioned to the process of installing numerous applications, making it easier for attackers to social engineer the user into installing a malicious application. It may be preferable for an attacker to introduce malicious code through a malicious application, rather than attempting to exploit a vulnerability in an existing application. Therefore, for platforms that do not provide isolation between applications, an increased importance is placed on the detection and mitigation of potentially malicious applications.

3. MOBILE SECURITY MODELS

Beyond exploring general security challenges with mobile environments, it is productive to look in-depth at the security models that real-world platforms employ. Better understanding the strengths and weaknesses of a platform's security model allows researchers to target their efforts towards specific platform deficiencies or against certain classes of threats. It is also beneficial to understand the development decisions and trade-offs made by the platform designers when attempting to balance security with usability and extensibility.

Unfortunately, mobile platforms have a diverse set of security models. No two platforms are the same when it comes to security mechanisms and design decisions, making the development of platform-agnostic protection mechanisms a significant challenge.

In order to understand the unique differences between the security models of various mobile platforms, we detail a simple taxonomy of common attributes for mobile security models. Our taxonomy decomposes the security of the mobile device platform into three primary components: application delivery, trust levels, and system isolation. For each component of the taxonomy, we rate the protection capabilities of the top 5 most popular mobile software platforms, as seen in Table 1. Such analysis may also be applied to other plat-

forms that are gaining in popularity such as Palm's webOS, Intel's Moblin, and OpenMoko.

3.1 Application Delivery

Application delivery refers ability of a mobile platform to verify the integrity of the source of an application. Secure application delivery capabilities are important to not only assert the source and identity of a particular application, but also to make it more difficult for an attacker to install a malicious application on a victim's device. However, it is a significant challenge for platform vendors to balance restrictive application delivery capabilities while maintaining sufficient extensibility of the mobile device. Numerous platforms offer the capability for applications to be cryptographically signed, assuring the end user of the identity of the application's developer. Obtaining signing keys from the platform's vendor may vary in difficulty and cost. Platforms may also lock down the mobile device and only allow installation of application from a single source. On the other hand, some platforms may choose to focus on open extensibility and allow applications to come from any source or developer.

For example, we classify the iPhone with "high" application delivery capabilities because each new application must be authenticated and go through an approval process performed by Apple before being published in the App Store. Apple also has the capability to revoke applications from the App Store and maintains a remote "kill switch" that allows Apple to blacklist applications that may have already been installed on a device.

The Android platform is given a "medium" rating due to it's default setting to only allow applications through the official Android Marketplace. However, a prominent user-accessible option is available to allow the installation of applications from non-Marketplace sources. When this option is enabled, applications may be installed from any web site, greatly increasing the risk of a user being tricked into installing a malicious application.

The RIM Blackberry platform is rated at "low" for application delivery. The platform supports signed applications and has a user-accessible option to allow unsigned applications, similar to the Android platform. However, signing keys can be purchased with anonymous prepaid credit cards by malicious parties from Blackberry for only $20. Therefore, Blackberry's signing capabilities may actually induce a false sense of security into users given the low bar for an attacker to create signed malicious application.

3.2 Trust Levels

Trust levels refer to the capability to assign a particular confidence or privilege to an application. Comprehensive trust levels are important to prevent applications from performing actions that they are not authorized to perform. These trust levels may be specified at numerous points in the application delivery and installation. Some platforms assigned a trust level when is signed by the vendor or developer. Cryptographic signatures may be used to determine whether an application is allowed to operate at an elevated trust level. Other platforms ask the user to decide what trust level an application may run at or present a set of desired privileges for the user to approve or deny. Choosing the optimal granularity of trust levels can present a challenge for mobile security models. If the trust levels are too coarse-grained, the risk of malicious behavior within applications may increase. If the trust levels are too fine-grained, it may raise performance concerns to track system events at such a low-level and usability concerns for users to be able to make an educated decision about an application's trust.

For example, Google's Android platform is rated at "high" as it has a permission-based model that strikes a good balance of trust level granularity. When an application is installed, a manifest provided with the application states the desired capabilities of the application (e.g., access the network, access the dialer, access coarse-grained location data). The user is prompted to review the requested capabilities and decide whether to allow the application to install.

The Windows Mobile platform is rated at "medium". While not as fine-grained as the Android platform, it provides three distinct tiers of permission: privileged, normal, and blocked. Privileged applications can perform any action they desire, normal applications are restricted to certain API calls and are denied access to certain system files, and blocked applications are completely denied execution.

The iPhone is rated at a "low" because it has very course grained permissions that only protect a few services such as the location of the user.

3.3 System Isolation

System isolation refers to the capability of the platform to isolate or sandbox a particular application and prevent it from compromising or affecting the underlying system or other applications. As vulnerabilities in complex mobile applications are not uncommon, a modern mobile software platform should include mechanisms to reduce the risk of a compromise and safe-guard the integrity of the underlying system.

For example, the iPhone platform is rated at "low" for system isolation as many of the applications run at the same privilege level. Therefore, if a vulnerability exists in such an application, the integrity of other applications may be compromised as well. Given the large attack surface of complex Objective-C based applications, the lack of system-wide sandbox functionality is cause for concern.

On the other hand, the Android platform is rated at "high" for system isolation. While a vulnerability within an Android application may allow an attacker to steal data owned by that application (e.g., steal cookies by exploiting a browser), other applications and the underlying system is isolated from the compromise since each app is executed as a unique UID.

4. COMMANDMENTS FOR MOBILE SECURITY RESEARCH

In this last section, we present several rules to help guide future research in the area of mobile security. These suggestions are based on observations of existing research in the academic community and explore some of the common pitfalls researchers encounter. While these commandments are intended to be presented in a light-hearted manner, we feel that researchers should them into consideration at a high-level when approaching future mobile security research.

4.1 Thou shall take forward lessons.

It's vital to be well-versed in the security of traditional computing environments in order to take forward lessons and recognize subtle differences in mobile environments. For example, we've seen the success that technologies such as NX,

ASLR, and stack cookies have had against memory corruption and control flow hijacking vulnerabilities in traditional computing, and we should therefore apply them appropriately to mobile environments. Taking forward lessons in the emerging field of mobility is especially important to form a solid, secure foundation and ensure that security does not take a backseat to functionality, since security concerns can often be overlooked without immediately impacting the user experience.

4.2 Thou shall justify mobile adaptation.

While it's important to learn lessons from securing traditional computing, researchers should not blindly take forward techniques and apply them to mobile environments without justification. We commonly see mobile security papers proposing the same security techniques used for fixed computing without justifying why the technique is appropriate for mobile environments. The requirement of effective usability in mobile environments may place unique pressures on the adaptation of security techniques, specifically given the case of limited device resources.

4.3 Thou shall consider multiple platforms.

Given the diverse security models that we have discussed, it's important that mobile security research considers the variations between platforms and the impact they may have. While this doesn't imply that a research project must have full implementations for every mobile platform in existence, it simply means that discussion should be present that justifies whether a proposed approach applies or doesn't apply to other platforms and security models. Failure to consider the diversity of security models across multiple mobile platforms may severely limit the scope and value of the research.

4.4 Thou shall be cognizant of future uses.

A common pitfall of mobile security research is designing policy enforcement mechanisms around current usage models. For example, designing a malware detection heuristic on the assumption that automated SMS sending represents malicious activity may be misguided, as rich applications commonly utilize such functionality nowadays. While it's not possible to tell the future, researchers should observe trends in fixed computing as potential indicators of future mobile uses.

4.5 Thou shall be cognizant of future threats.

Similarly, researchers must be cognizant of potential future threats and attack techniques when approaching mobile security. Simply evaluating a particular scheme against the handful of existing mobile malware threats and calling it a day is inadequate. Most importantly, security mechanisms must scale elegantly in terms of resource consumption and adaptability as mobile threats increase in quantity and complexity in the future.

5. CONCLUSION

We expect consumer mobile platforms and devices to continue their rapid expansion in terms of sophistication and functionality. As their popularity increases and they become enticing targets for attackers, these devices will face a range of new security threats. We believe that domain of mobile security presents a number of interesting challenges that are becoming ever-important to explore as the adoption and use of these mobile platforms continues to accelerate.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation (NSF) under contract number CNS 0627445 and the Department of Homeland Security (DHS) under contract number NBCHC060090.

7. REFERENCES

[1] Personal Communications with the Google Android Security Team, 2009.

[2] L. Cox and P. Chen. Pocket hypervisors: Opportunities and challenges. *Proceedings of HotMobile*, 2007.

[3] D. Dagon, T. Martin, and T. Starner. Mobile phones as computing devices: The viruses are coming! *IEEE Pervasive Computing*, 2004.

[4] J. Ekberg and M. Kylánpää. Mobile Trusted Module (MTM)–An Introduction. *Nokia Research*, 2007.

[5] S. Furnell. Handheld hazards: The rise of malware on mobile devices. *Computer Fraud & Security*, 2005.

[6] T. Garfinkel and M. Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *10th Workshop on Hot Topics in Operating Systems*, 2005.

[7] J. Golic. Cryptanalysis of alleged A5 stream cipher. *Lecture Notes in Computer Science*, 1233:239–255, 1997.

[8] C. Mulliner and C. Miller. Fuzzing the phone in your phone. In *Proceedings of BlackHat USA 2009*, 2009.

[9] J. Oberheide, E. Cooke, and F. Jahanian. CloudAV: N-Version Antivirus in the Network Cloud. In *Proceedings of the 17th USENIX Security Symposium*, San Jose, CA, July 2008.

[10] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian. Virtualized In-Cloud Security Services for Mobile Devices. In *Workshop on Virtualization in Mobile Computing (MobiVirt '08)*, Breckenridge, Colorado, June 2008.

[11] A. O'Donnell. When malware attacks (anything but windows). *IEEE SECURITY & PRIVACY*, pages 68–70, 2008.

[12] Sergio Alvarez. The smart-phone nightmare. `http://cansecwest.com/csw09/csw09-alvarez.pdf`, 2009.

[13] R. Sharp, A. Madhavapeddy, R. Want, and T. Pering. Enhancing web browsing security on public terminals using mobile composition. 2008.

[14] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, T. La Porta, and P. Mcdaniel. On cellular botnets: Measuring the impact of malicious devices on a cellular network core. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, 2009.

[15] L. van Doorn. Trusted computing challenges. In *STC '07: Proceedings of the 2007 ACM workshop on Scalable trusted computing*, 2007.

[16] L. Xie, X. Zhang, A. Chaugule, T. Jaeger, and S. Zhu. Designing System-level Defenses against Cellphone Malware.