The Challenges of Enterprise Mobility
IQT Quarterly is a publication of In-Q-Tel, Inc., the strategic investment firm that serves as a bridge between the U.S. Intelligence Community and venture-backed startup firms on the leading edge of technological innovation. IQT Quarterly advances the situational awareness component of the IQT mission, serving as a platform to debut, discuss, and debate issues of innovation in the areas of overlap between commercial potential and U.S. Intelligence Community needs. For comments or questions regarding IQT or this document, please visit www.iqt.org, write to iqtquarterly@iqt.org, or call 703-268-3000. The views expressed are those of the authors in their personal capacities and do not necessarily reflect the opinion of IQT, their employers, or the Government.

©2012 In-Q-Tel, Inc. This document was prepared by In-Q-Tel, Inc., with Government funding [U.S. Government Contract No. 2009*0674524*000]. The Government has Government Purpose License Rights in this document. Subject to those rights, the reproduction, display, or distribution of the Quarterly without prior written consent from IQT is prohibited.

EDITORIAL

IQT Quarterly, published by In-Q-Tel, Inc.

Editor-in-Chief: Lisa L. Bader

Editor: Emma Resnick

Theme Editor: Jay Emmanuel

Contributing Editors: Brittany Smith and Michelle Higgins

Managing Editor: Lisbeth Poulos

Design by Lomangino Studio, Inc.

Printed in the United States of America
# Table of Contents

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Our Radar: The Challenges of Enterprise Mobility</td>
<td>02</td>
</tr>
<tr>
<td>By Jay Emmanuel</td>
<td></td>
</tr>
<tr>
<td>A Look Inside: Slide to Unlock</td>
<td>04</td>
</tr>
<tr>
<td>Moving Towards a Capability Package for Secure Mobile Virtualization</td>
<td>05</td>
</tr>
<tr>
<td>By David Kleidermacher</td>
<td></td>
</tr>
<tr>
<td>Cellular Network Security and Control</td>
<td>09</td>
</tr>
<tr>
<td>By Ben Weintraub</td>
<td></td>
</tr>
<tr>
<td>The Greatest Risks in Mobile Apps</td>
<td>14</td>
</tr>
<tr>
<td>By Anthony Bettini</td>
<td></td>
</tr>
<tr>
<td>Connecting Identity and Mobility: A Secure, Scorable, and Sustainable Mobile Wallet Approach</td>
<td>18</td>
</tr>
<tr>
<td>By Siva Narendra</td>
<td></td>
</tr>
<tr>
<td>Time to Make the Apps</td>
<td>22</td>
</tr>
<tr>
<td>By Marshall Vale and Andrew Yu</td>
<td></td>
</tr>
<tr>
<td>Tech Corner</td>
<td>26</td>
</tr>
<tr>
<td>A technology overview from IQT portfolio company Mocana</td>
<td></td>
</tr>
<tr>
<td>In the News</td>
<td>29</td>
</tr>
</tbody>
</table>
THE CHALLENGES OF ENTERPRISE MOBILITY

By Jay Emmanuel

Welcome to the Summer 2012 issue of the IQT Quarterly. With a special focus around the challenges of enterprise mobility, this issue builds on topics discussed in our Winter 2011 edition, “Moving to Mobile: Trends, Technology, and Solutions.” Since the publication of that issue, IQT has continued to collaborate with our Intelligence Community partners to brainstorm architectures and designs for secure and scalable enterprise mobile rollouts. This has been a tremendous learning experience for IQT as we cultivate our understanding of the IC’s requirements and track trends in the rapidly evolving commercial market. These discussions have allowed us to focus on companies and technologies that address current government challenges.

Mobility for the enterprise is an inevitable phenomenon that will continue to grow over the next few years. The Blackberry-centered mobile framework provided a one stop, locked down solution for IT organizations that was easy to adopt and use. With RIM fast imploding and with the tremendous consumer uptake of Android and iOS devices, the mobile IT environment has become far more diverse and challenging in terms of devices, operating systems, security risks, and other issues that are commonly of concern to organizations. Users within the enterprise now demand a mobile experience that is on par with or better than commercial applications. This new mobile world is redefining the function of traditional IT in the workplace and presents organizations with a set of complex security, management, cost, compliance, and legal issues that need to be addressed.

Security

Mobile devices provide a much larger and more diverse attack surface for an adversary. With the growing trend of Bring Your Own Device (BYOD) policies in organizations, these risks are further elevated. A lost or stolen employee-owned device with permanent VPN connections for email and other applications can provide easy access into the enterprise network where corporate data and sensitive information are stored. Techniques like device locking, data-at-rest encryption, enterprise- and application-level sandboxing, device virtualization, and secure boot loaders that mitigate corporate risk are all evolving in the commercial world. Mobile Device Management (MDM) and Mobile Application Management (MAM) are relatively mature and help with risk mitigation — it is not expensive or difficult to get robust and scalable tools that are designed for this next generation of mobile IT.
Policy and Compliance

Establishing and adhering to an organization-wide policy for mobile devices is a key step towards the effective management of enterprise technology. Questions like whether or not to allow BYOD, how to effectively sandbox user and personal data, how much to lock down an enterprise device, whether or not to allow split tunneling of user traffic, and whether to route all uplink and downlink traffic through enterprise infrastructure need to be addressed at both a corporate and department level since all have ramifications at various levels within the organization. Once organizational policies are defined clearly, employee education and enforcement of the policy on the device and network are necessary. Plans for immediate and effective remediation should be implemented in the event a policy is violated.

Usability

Consumers have grown to expect a user experience from enterprise applications that matches or exceeds that provided by commercial applications. A clunky enterprise app or a device that is considered too locked down isn’t likely to be widely adopted, and users will find ways around policy that is deemed to be too restrictive. Optimizing usability requires that the enterprise create commercial quality apps without compromising the security posture of the device or the network. Consumers are comfortable downloading apps from an app store; creating a similar experience in the enterprise will likely increase enterprise application use and adoption. Users should also be confident that they are free to use their devices for personal use with the assurance that there will be a clear separation between personal and enterprise content.

Enterprise Applications

Enterprise mobile applications need to effectively extend backend IT services and present enterprise data to a mobile device quickly and easily. Integration between the mobile device and backend systems should be a seamless and effective process both for an enterprise app developer and the end user. For the developer, implementing security policies and access to backend services like authentication, directory, and authorization services should be a transparent process that is easily incorporated using standard libraries and APIs. Given the wide array of devices and operating systems that the application needs to work on, solutions that allow web developers with traditional programming skills to create device and operating system agnostic mobile applications will be critical in achieving application development scale.

Rapid Rate of Change

Designing a scalable and secure mobile enterprise architecture is complex, especially given the fact that the technology platforms and devices are evolving at a fast pace. Most IT organizations are accustomed to a Microsoft-like 5-6 year product lifecycle in which they have the time to plan changes and slowly implement them. Mobile evolves very fast from every perspective — devices, versions of operating systems, diversity of available applications, and the types of security threats on these devices are constantly changing. A well thought-out and successful mobile strategy will plan for change and select architectures that allow for future adaptation to emerging trends.

Cost Management

A mobile device’s cost continues long after initial deployment and is often challenging to determine and manage. Besides the capital costs associated with the devices, usage rates are only increasing with the proliferation of bandwidth-hungry applications. With BYOD, enterprise users present the organization with a wide array of bills and user-purchased service agreements. Cost visibility in real time is essential to defining and maintaining expense policies. International roaming charges can have a serious impact on costs and have been the focus of many recent cost-related mobile decisions. Most Mobile Device Management suites provide real-time cost visibility. Targeted analytics that yield insight into usage patterns in relation to business operations and geographies can help with understanding costs.

These are but a few of the many challenges that are associated with mobile IT. IQT will continue to engage with cutting-edge companies in this space to stay abreast of technology trends. While this is sometimes a daunting task since this is a relatively new and emerging area of technology, it remains exciting and challenging to us as we help guide customers on strategies in the mobile space.

Jay Emmanuel serves as Vice President in IQT’s Information and Communication Technologies Practice. Prior to joining IQT, he worked extensively in the mobile space with Motorola, Hughes Network Systems, and most recently with Megisto Systems, a startup that developed carrier-grade mobile gateways. Jay has a Master of Science in Computer Engineering from the University of Maryland at College Park.
A LOOK INSIDE: SLIDE TO UNLOCK

This edition of the IQT Quarterly considers the ever-expanding realm of enterprise mobility. This expansion is centered around innovative companies exploring how to secure mobile devices and maximize their use in the enterprise.

The issue opens with an article by David Kleidermacher discussing the security challenges of mobile devices, which may be enterprise-owned or personal. He also considers how mobile virtualization — the use of a hypervisor to protect functions performed in trusted software outside of the mobile operating system — could be used for highly secure operating environments.

Ben Weintraub follows with an analysis of Mobile Virtual Network Operators (MVNOs), which purchase access to the networks of facilities-based carriers and rebrand and resell the network services to their own customers. MVNOs enable organizations to implement unique security and control features.

By showing how software procurement processes and operating systems running on mobile devices are drastically different from PC architectures, contributor Anthony Bettini explains how the risks and potential solutions in mobile security are equally different from their predecessors. He also classifies various intentional behaviors in mobile apps that pose a risk to the enterprise and its data.

Next, Siva Narendra discusses the budding potential of mobile wallets. The convenience of being able to access all identities via an “always-on” and “always-with-you” mobile device is intriguing, but poses security risks to the identities and associated data. Narendra considers how to maintain the convenience and sustainability of mobile wallets without compromising security.

In their article, Marshall Vale and Andrew Yu discuss the evolution from classic desktop applications to mobile apps, and how their distinct characteristics can teach us to optimize mobile apps for the enterprise. The authors note that successful mobile apps offer select functionality to make them appropriate in the mobile context, and that an increasing number of business apps are written internally by the end users.

This issue of the IQT Quarterly is the beginning of a conversation about how to optimize the use of mobile devices within the enterprise while securing organizational data and assets. As organizations embrace the functionality and convenience of mobile operations, IQT will continue to seek out innovative technologies that address the user needs and security challenges of enterprise mobility.
Moving Towards a Capability Package for Secure Mobile Virtualization

By David Kleidermacher

The enterprise world is going mobile, driven by distributed workforces, the need to stay connected, and the sheer productive horsepower of modern handheld devices. However, unlike PCs that are traditionally IT owned and managed, handhelds are inherently personal: we use them to watch movies, browse photos, chat with family and friends, and play games.

Enterprises cannot tolerate the use of these unmanaged personal devices for the processing of sensitive information and connection to internal corporate networks. The inexorable consumerization of IT demands a solution to Enterprise-Owned Device (EOD) and Bring Your Own Device (BYOD), where an employee can use a single handheld for personal activities or in the workplace while ensuring both parties are satisfied with security, usability, and cost.

One promising response to the EOD/BYOD trend is mobile virtualization, where a hypervisor is retrofitted below the primary mobile operating system. Critical security functions, such as user authentication and data-at-rest encryption, can be performed in trusted software that resides outside of the mobile OS and is protected and isolated by the hypervisor, enabling high assurance protection of enterprise information.

Meanwhile, the U.S. Intelligence Community and DoD are aggressively looking for ways to take advantage of commercial off-the-shelf (COTS) handhelds in sensitive environments without incurring the cost and time-to-market challenges of the traditional NSA Type-1 cryptographic process. This desire led the NSA to establish the Commercial Solutions for Classified (CSfC) Program, which is defining capability packages that provide requirements guidance for leveraging commercial products for classified environments. These packages reference a suite of emerging Common Criteria protection profiles, which define functional security requirements for important classes of software and systems.

Neither a capability package nor a related Common Criteria protection file yet exists for mobile virtualization. This article provides an overview of the factors that will influence these standards, including security challenges facing modern mobile operating systems and how mobile virtualization can be used to implement government-approved layered encryption approaches.

Android Security Retrospective

Android is the most popular mobile OS. In addition, the U.S. government has favored Android in its mobility initiatives due to the relative openness and ease of customization. Therefore, as we look at the security posture facing modern mobile operating systems, we focus on Android, although the problems and conclusions are by no means specific to it.

Shortly after the first Android phone was released in 2008, researchers discovered that Android would silently and invisibly execute all commands typed into the device with superuser privileges. A ZDnet writer called this vulnerability the "Worst. Bug. Ever."1

In late 2010, security researchers uploaded to the Android market a spoofed Angry Birds game application that invisibly downloaded other apps without the user’s approval or knowledge. The extra downloads were malicious, stealing the phone’s location information and contacts, and sending illicit text messages. The researchers reported numerous weaknesses in Android, including a faulty use of SSL, a lack of application authentication, an easy method of breaking out of the Android Dalvik virtual machine...
sandbox via native code, and the focus of the attack—a weak permissions architecture.2

Next, we visit the U.S. CERT National Vulnerability Database. A search on Android turns up numerous vulnerabilities of varying severity. Here is a sampling of the worst offenders (remotely exploitable):

- **CVE-2011-0680**: Allows remote attackers to read SMS messages intended for other recipients
- **CVE-2010-1807**: Allows remote attackers to execute arbitrary code
- **CVE-2009-2999, -2656**: Allows remote attackers to cause a denial of service (application restart and network disconnection)
- **CVE-2009-1754**: Allows remote attackers to access application data
- **CVE-2009-0985, -0986**: Buffer overflows allow remote attackers to execute arbitrary code

Android is, of course, susceptible to Linux kernel vulnerabilities as well. The rapid development and monolithic architecture of Linux has been well publicized. Linux kernel authors have published multiple installments of a Linux development statistical overview, and the numbers are staggering.3 With 20,000 lines of kernel code modified per day, 6,000 unique authors, and rapid growth (now 15 million lines of code), it should come as no surprise that dozens of Linux kernel vulnerabilities are reported each year. While a significant portion of the churn in the code base is due to support for new microprocessors and peripherals, the core kernel, including networking and file system support, also undergoes rapid change. CVE-2009-2692, informally known as the proto-ops flaw, is a set of bugs in the Linux kernel’s management of file and network access objects. A trivial user mode program can be used to subvert an Android system using this vulnerability. The proto-ops flaw was latent in the Linux kernel for eight years before researchers discovered it.

Yes, we paint a grim picture of Android security. However, the picture is based on simple facts that should not be surprising: Android was never designed to provide a high assurance of security. Numerous Linux/Android “hardening” approaches, such as SE Android and grsecurity, can prevent many attacks and should be used when practical, but they cannot change the overall assurance of Android and will not defeat sophisticated attackers.

**Mobile Phone Data Protection: A Case Study of Defense-in-Depth**

Mobile device popularity juxtaposed with its lack of strong security has sparked a rigorous scramble by software vendors, device OEMs, systems integrators, and government security evaluators to find ways to retrofit improved system security to commercial devices. One approach to raising the level of assurance in data protection within an Android-based device is to employ multiple encryption layers. Indeed, this defense-in-depth approach is promoted by the NSA mobility capability package. For example, an Android smartphone can use a layer four (OSI model) SSL VPN client to establish a protected data communication session. An Internet Protocol Security (IPsec) VPN application, running at layer three, can be used to create a second, independent connection between the smartphone and the remote endpoint (Figure 1). This secondary connection uses independent public keys to represent the static identities of the endpoints. The data in transit is doubly encrypted within these two concurrent connections. This layered security approach is an example of defense-in-depth.

The concept of defense-in-depth originates in the military: multiple layers of defense, such as a combination of mines and barbed wire, rather than just mines or barbed wire alone, are used to increase the probability of a successful defense as well as to potentially slow the progress of an attacker. Defense-in-depth has been successfully applied in war since ancient times, and the concept is alive and well in the information security age.

Layered SSL/IPsec data protection is a sensible application of defense-in-depth to counter threats. If an attacker is able to break the SSL encryption, the IPsec layer will continue to protect the data. An attacker may be able to steal the SSL keys but not the IPsec keys. The attacker may be able to install malware into the SSL application but not the IPsec application. The SSL application may exhibit side effects and the IPsec application can mitigate those attacks.

![Figure 1](image_url) | Multiple layers of encryption within Android
channel weaknesses to which the IPsec application is immune. To succeed, the attacker must break both the SSL and IPsec encryption layers.

Clearly, this layered approach depends on the independence of the layers. Most importantly, the SSL and IPsec private keys must be independently stored and immune to a single point-of-failure compromise. However, in a typical mobile environment, both the SSL and IPsec long-term private keys are stored within the same file system and managed by a vulnerable mobile OS. Compromise does not require attack sophistication; a single Android root vulnerability can expose both layers.

The run-time environment must provide strong isolation of the SSL and IPsec application layers, and must not provide an attack surface through which to break that isolation. The mobile OS itself is clearly unable to provide a strong isolation due to vulnerabilities and a weak privilege model. Thus, we turn to retrofitting a mobile virtualization layer below the mobile OS.

In addition to its potential for improved security, mobile virtualization enables use cases that are simply not possible otherwise. For example, mobile virtualization enables dual persona, in which two independent instances of mobile operating systems (perhaps even different versions) are hosted within the same handheld. One instance may be used for connecting to unclassified networks or the Internet, while another instance may be managed by the enterprise, accessing corporate or classified networks.

Let’s take a look at the major types of mobile virtualization architectures in the context of the layered encryption goal.

**Containers**

Linux has a concept of containers called LXC. Containers are not a form of system virtualization. Rather, containers implement what is called OS virtualization, providing execution environments with access to a subset of the available file system and sometimes separate CPU scheduling resources. Containers can provide the illusion of multiple personas running, for example, two instances of Android and/or other software environments.

This approach can be used to implement the layered encryption use case: one container can host an SSL connection that wraps communications from the underlying Linux and its native IPsec. However, both layers depend on the security of the single mobile OS kernel.

**Type-2 Hypervisor**

Type-2 hypervisors are similar to containers in that the secondary environment runs as an application on top of the primary OS. However, instead of hosting only a private file system and its contained applications, the secondary persona is a full-fledged guest OS running within a virtual machine created by the hypervisor application (Figure 2). The hypervisor uses the primary OS to handle input/output (I/O) and other resource management functions.

Type-2 mobile hypervisor products, such as VMware MVP and Linux KVM, are used to provide an enterprise management persona on top of the primary Android environment. The virtualized Android can use an SSL connection to the enterprise while the underlying Android’s IPsec is also used to wrap the communication between endpoints.

However, the Type-2 model fails to provide strong isolation. Faults or security vulnerabilities in the primary general-purpose OS will impact the critical functions running in the virtual machine. Furthermore, Type-2 hypervisor applications deployed in the enterprise space have themselves been found to contain vulnerabilities that break the sandbox.

**Type-1 Hypervisor**

Type-1 hypervisors also provide functional completeness and concurrent execution of a secondary persona. However, because the hypervisor runs on the bare metal, persona isolation cannot be violated by weaknesses in the mobile OS. Thus, a Type-1 hypervisor represents a promising approach from both a functionality and security perspective. But the hypervisor vulnerability threat still exists, and not all Type-1 hypervisors are designed to meet high levels of security.
One particular variant, the microkernel-based Type-1 hypervisor, is specifically designed to meet high assurance, security-critical requirements. Microkernels are well known to provide a superior architecture for security relative to large, general-purpose operating systems such as Linux and Android.

In a microkernel Type-1 hypervisor, system virtualization is built as a service on the microkernel. Thus, in addition to isolated virtual machines, the microkernel provides an open standard interface for lightweight critical applications that cannot be trusted to a general-purpose guest. For example, SSL can be hosted as a microkernel application or in a separate Linux virtual appliance, providing the highest possible level of assurance of isolation between the encryption layers. IPsec packets originating from Android are doubly encrypted with the SSL service before transmission over the wireless interface (Figure 3).

The isolation properties of some secure microkernels can even protect against sophisticated covert and side-channel software-borne attacks. The microkernel can also manage and utilize a hardware root-of-trust, such as a smartcard microcontroller embedded within a microSD or SIM card when available, providing protection against physical attacks on critical data, such as cryptographic keys (Figure 4). One example of microkernel Type-1 hypervisor is the INTEGRITY Multivisor from Green Hills Software. The Multivisor’s microkernel is widely deployed in enterprise and embedded electronics and NSA-certified cryptographic devices, and is the only software technology certified to Common Criteria EAL 6+ / High Robustness, the level required to protect high-value information against sophisticated attackers.

Layered encryption as a defense-in-depth strategy is a sensible approach to increasing the assurance of mobile device data protection. However, what is not sensible is to entrust the mobile OS with both layers. There is simply too much vulnerability to prevent the layers from being subverted. Modern microprocessors, only recently imbued with hardware virtualization acceleration features, and mobile hypervisor solutions provide the requisite features to get the best of both worlds: the power of mobile multimedia and applications deployment infrastructure alongside, but securely separated from, critical system security functions.

**Figure 3** | Microkernel Type-1 hypervisor approach to layered data-in-motion encryption

**Figure 4** | Adding physical security protection via attached smartcard to the microkernel Type-1 hypervisor

---

**David Kleidermacher** is Chief Technology Officer at Green Hills Software, where he is responsible for technology strategy, platform planning, and solutions design. He is a leading authority in systems software and security, including secure operating systems, virtualization technology, and the application of high robustness security engineering principles to solve computing infrastructure problems. David earned his Bachelor of Science in computer science from Cornell University and has been with Green Hills Software since 1991.

**REFERENCES**

2. Jon Oberheide and Zach Lanier; TEAM JOCH vs. Android; presentation made at ShmooCon 2011 conference.
CELLULAR NETWORK SECURITY AND CONTROL

By Ben Weintraub

As the ratio of active wireless units to the population count in the U.S. jumps over 100 percent, Mobile Virtual Network Operators (MVNOs) have arisen to serve market niches overlooked or underserved by facilities-based operators. MVNOs now provide approximately 10 percent of the wireless devices in the U.S., serving the needs of specific niches such as healthcare, education, prepaid services for the underprivileged, select ethnic communities, and the elderly. An MVNO can leverage a wide range of flexibility in technical and business arrangements to provide custom wireless solutions in a highly capital-efficient manner.

What is an MVNO?

An MVNO purchases access to one or more facilities-based carriers’ networks and resells the network services to customers under its own brand. An MVNO owns the relationship with its customers and performs marketing, distribution, billing, and customer care independent of the underlying carrier. Some MVNOs go further and provide technologically differentiated services in order to satisfy the needs of a customer niche.

As of 2010, approximately 60 MVNOs were operating in the U.S. and that number is likely much larger today. The MVNO model has proven successful in many countries, with scores of MVNOs operating outside the U.S. Companies often pursue an MVNO model to create a branded wireless experience where the brand is at the forefront. One such company is i wireless, offering traditional wireless service [e.g., voice, text, and data] to customers of American supermarket chain Kroger. The company has integrated the retail point of sale system with the wireless billing system to offer loyalty rewards.

A company operating as an MVNO has a wide range of flexibility to design not just the rate plans and purchase experience, but also to utilize its own network equipment.
to control cellular traffic at a very granular level. For example, kajeet is an MVNO serving the youth market with parental controls. The service leverages its own mobile service control point and data environment to regulate every voice call, text message, or packet of data going to or from the mobile device. The kajeet platform leverages other features typically associated only with facilities-based carriers such as network-based location services, custom Domain Name System (DNS) resolution, and the ability to control over-the-air device updates.

The availability of inexpensive mobile data devices, coupled with the need for ubiquitous connectivity has created opportunities for many new MVNOs. These MVNOs focus on creating an “Internet of Things,” such as connected cars (e.g., Tesla, Ford Sync), e-readers, and security and healthcare-related devices.

All of the major carriers in the U.S. offer wholesale agreements. The range of services included in such wholesale agreements and the pricing vary from carrier to carrier. The commercial arrangement typically includes setup fees, a la carte usage fees for minutes, texts, and megabytes, and monthly minimums. Other miscellaneous fees may apply for number porting, picture messaging, location services, etc. Typically, usage rates are low enough to enable the MVNO to create competitive retail offers.

However, the more interesting opportunity for MVNOs is to bring unique solutions to problems in the market. Carrier wholesale relationships vary widely in regards to the technical integration they allow from an MVNO. Historically, Sprint has been the most open network. For example, Sprint allows deep network integration such as MVNO-hosted service control points, Home Agents (HAs), and Authentication, Authorization, and Accounting protocols (AAAs).

**MVNEs and MVNAs**

While many MVNOs establish their own business support systems, operations support systems, staffing, and procedures to run their back office, others turn to enablers. Mobile Virtual Network Enablers (MVNEs) are companies that assist MVNOs in launching and operating their services. The definition of an MVNE varies widely and includes companies that provide components of the solution, such as Telcordia’s Managed-Hosted Service Control Point, or billing software vendors such as Qualution.

Other MVNEs, such as Arterra Mobility (a division of kajeet, Inc.), provide an integrated solution for billing, care, logistics, etc. MVNEs leverage experience, staff, and sometimes multi-tenant network equipment and software platforms across several installations and allow the MVNO to stay focused on their business objectives, particularly the design of differentiated solutions and customer acquisition. An MVNE can also assist or manage regulatory compliance for an MVNO such as FCC registrations, universal service fees, and taxes.

A Mobile Virtual Network Aggregator (MVNA) takes the MVNE model one step further. An MVNA aggregates its clients’ wholesale usage into one carrier contract. This typically improves the launch timeline and costs by eliminating the load the facilities-based carrier must bear (since the MVNA is already setup and integrated). It can also enable MVNOs to take advantage of lower network service costs than those they would pay were they to purchase such services directly from the carrier.

**An Architecture for Security and Control**

Traditionally, MVNOs have been focused on marketing and pricing strategies, while the host carrier has provided all of the core telecom functions, including the routing and transporting of traffic. Over the past few years, however, carriers have opened their networks to allow MVNOs to control or replace some of the host carrier’s network functions.

All carriers now allow MVNOs a variety of network configurations, which allow the MVNOs to secure and control their traffic. Some carriers allow MVNOs deep access to core network functions including hosting their own HAs, AAAs, and voice switches. A host carrier always provides the radio-access network, which includes the spectrum, towers, base-tower-subassembly, switches, and Packet Data Serving Node (PDSN). Figure 1 illustrates the type of network access available with one of the major carriers. Voice, text, and data traffic are handled by different components in the host carrier network.

The configuration in Figure 1 includes an MVNO-operated service control point (SCP) that interacts with
the host carrier’s mobile switching center (MSC). With access to SS7 signaling, the SCP can allow, disallow, or reroute every inbound and outbound voice call. The signaling information contains a variety of data including the origin and destination number and local switch time. The SCP can be programmed to handle those calls according to the MVNO’s business logic. This allows the complete blocking of voice calls (except for 911), allowing calls only to or from specific numbers, applying time-of-day rules, or routing of outbound calls to specific destinations. For example, outbound calls, regardless of the dialed number, could be redirected to headquarters. The text messaging architecture outlined in Figure 1 mimics the voice capabilities.

Alternately, more secure architectures are possible. Some host carriers allow the voice-bearer traffic to be transferred to the MVNO’s private network from the MSC. Similar architectures can be arranged for text messaging.

The data architecture illustrated in Figure 1 includes the routing of bearer traffic, in addition to signaling traffic, into the MVNO’s network. This configuration is based upon Sprint’s Datalink or Verizon’s Private Network architectures and utilizes dedicated Multiprotocol Label Switching (MPLS) connections to securely route data traffic from the carrier’s network to the MVNO.

In this design, cellular traffic from the smartphone or tablet is directed to a private, secure MVNO-managed wireline network that enables a broad spectrum of varying security and control solutions based on predetermined or dynamic policies. Once within the MVNO environment, the data can be restricted to a private network, or filtered based on content before it hits the public Internet. Traffic can also be monitored for malware and rogue software on the device. All of the routing from the phone to the MVNO core is obfuscated: Core MVNO routers do not have public IP addresses nor do they route traffic over or through the Internet, which, for example, makes Denial of Service attacks (DoS) an unrealistic possibility. Similarly, third parties cannot initiate a direct connection to any of the smartphones or tablets. By filtering and monitoring
data traffic, it is also possible to control over-the-air device software updates, which may pose security risks such as attempts to disable third-party client software.

The MPLS network also allows the ability to create separate labels for independently secure zones over the same network. For example, one group of smartphones in a “green” zone can utilize the same physical connections but not have any access to traffic from a second group of smartphones in a “red” zone even if both “green” and “red” traffic coexists and traverses the same routers and ports. This is possible because MPLS is actually a VPN. MPLS VPN is connectionless, unlike traditional Internet Protocol Security (IPsec) VPNs, which require complex point-to-point mesh topologies that are difficult to scale. If desired, an additional layer of security may be added by utilizing IPsec on top of MPLS VPNs in the core, or IPsec can be used from the client device to its destination.

Because the signaling traffic is available as events arise, it is possible to monitor activity with extremely high granularity and low latency. Business rules can be constructed to analyze activity and take appropriate actions. For example, alerts can be designed to identify when a phone call is made to a particular number, or when users have exceeded a consumption threshold. A business rule can be established to perform a location request each time a call or data session is initiated.

As another example, it would be possible to match device use against a software client’s “heartbeat.” A heartbeat message sent from the client software to an operator is an indicator that client software (e.g., encryption software) is working. Lack of a pulse could mean several things: the phone is off, out of network coverage, or the client software is not functioning. However, as an MVNO, an operator can compare real-time device traffic against heartbeat messages (or lack thereof). If cellular activity (e.g., voice, text, or data traffic) is detected without corresponding messages that the client is “alive,” the device can immediately be located, disabled, calls can be rerouted, and alerts can be issued.

Of course, the security of signaling and bearer traffic is always dependent upon the host-carrier networks. Traffic to and from the devices must flow over the host carrier’s network from the radio access network to the serving MSC/PDSN until the traffic can be transferred to the MVNO’s network and facilities. Tower-to-mobile device security varies by network type and this must be evaluated carefully in the overall security posture.

In roaming situations, some of the routing and control scenarios break down. In the case of the kajeet consumer service, roaming has been disabled to ensure that all traffic, of any type, is fully controlled. However, the obvious disadvantage is that the end-user device only has access to the wholesale provider’s organic radio access network. If roaming is desired, the impact on security and control must be evaluated on a case-by-case basis.

Other connections to the host-carrier network provide access to basic operational functions such as provisioning, account management, troubleshooting,
number porting, picture messaging, and location-based services (LBS).

Network-initiated LBS implementations vary by carrier. In the case of Sprint, locations are performed via assisted GPS (AGPS), initiated over the control-plane, and do not rely on a handset client. Responses from the network include the location of the device and accuracy estimates. AGPS accuracy can be quite high where full GPS reception is available, or low where only tower reception information is available. AGPS locates can also be faster than GPS locates as the GPS is seeded with the approximate position of the device from the radio access network.

The relative robustness of AGPS over client-based GPS can facilitate interesting use cases. For example, credit card companies are contemplating use of LBS to verify that the purchaser (at least his or her phone) is in the same location as the retail location attempting to charge the card.

**End-User Devices and Software Clients**

Network-based security can complement client-level security as part of a layered approach or work standalone.

End-user device operating systems do not typically allow client software to granularly control the telecom functions of the device. For example, in order to provide control over voice, text, cellular data, and Wi-Fi on an Android device, client software must be developed at the kernel level. This requires software to be ported from device to device, slowing deployment times when a new device is released. Some devices, such as MiFis or embedded modem devices, do not support sophisticated software clients. Additionally, carrier issued OS updates may nullify kernel-level modifications. An MVNO architecture can mitigate or eliminate these gaps.

Network-based routing and controls are universal: they will control traffic over that network regardless of the device or the presence of client software. However, there are other areas where client software is required. The MVNO architecture does not inherently control access to Wi-Fi networks or USB connections. Additionally, functions that do not require network access or applications on the device can only be controlled by client software. Layering client-based encryption on top of a secure network can mitigate gaps in the host carrier network or roaming partners.

An MVNO architecture can provide organizations with security and control features not available through client-only implementations. Additionally, in-network security allows greater flexibility in device selection and deployment. The MVNO model is a prominent, international business model and extensions of the architecture described above can be utilized to work across multiple carriers simultaneously, either through MVNO or roaming agreements.

---

**Ben Weintraub** is the COO and founder of kajeet, Inc., a wireless service provider dedicated to children and education. Kajeet operates the Arterra MVNE/MVNA platform, enabling third parties to launch customized, branded wireless solutions. Ben is responsible for marketing, sales, and operations, overseeing the company’s strategic direction and the architecture and engineering infrastructure that power Arterra. He is an inventor on four of the company’s patents and several others that are pending. Prior to kajeet, Ben was the founder and COO of simplywireless.com. He also spent 6 years in the U.S. Navy, including a term as an Assistant Program Manager in the nuclear surface ship propulsion program. He earned a B.S. and M.S. in engineering from MIT, and a Masters of Engineering in nuclear reactor design through the Department of Energy.

**REFERENCES**

4. [https://www.wirelesshome.com/](https://www.wirelesshome.com/)
THE GREATEST RISKS IN MOBILE APPS

By Anthony Bettini

Due to the lack of advanced protective mechanisms such as sandboxing and broad use of code signature enforcement by the Microsoft Windows operating system, vulnerabilities accidentally inserted into code by programmers have traditionally presented one of the greatest information security-related risks within the enterprise. In contrast, the greatest risks associated with mobile apps are not related to insecure programming mistakes, but rather to the intentional misuse of supported features, which put the enterprise and sensitive data at risk.

Software is an enabler. Software has allowed new businesses to be successful, governments to run more efficiently, and people to improve their lives. But software has also introduced new risks to all of these parties and counter-parties. The ability to measure risk from software has evolved significantly in the past decade. Ten years ago, when looking at software vulnerabilities or “threats,” people in the commercial sector would primarily focus on the absolute impact from the worst possible outcome associated with exploiting the vulnerability.

While focusing on absolute impact is still common, times have changed, and more researchers are regularly focusing on attacks first. By better understanding attacks, we can understand where our resources are best focused to mitigate the risk of an attack succeeding and to mitigate the damage done when an attack does succeed. As all systems of importance will be attacked, the architecture of the system appears to have a strong causal effect on the attacks that occur against that system.

Mobile is Different

To understand why mobile is different from the security perspective, it is helpful to refresh our understanding of security for the Microsoft Windows platform. The vast majority of commercial sector security solutions have been designed around improving the security for Microsoft Windows deployments. Whether the product has been a “next-generation enterprise firewall” or an anti-virus engine, it was likely built with the goal of better protecting Microsoft Windows desktops and servers. Security products have been built this way because within the enterprise, desktop and server deployments have traditionally been almost entirely Windows-based. Use
of Windows in the consumer space has been similarly popular, with more than 90 percent of consumers using Windows as their desktop OS. Prevalence has therefore dictated the focus, both for attackers as well as for those charged with defending systems and networks.

**OS Architecture**

Fast-forward to the present, where mobile is being adopted at a fast pace and mobile devices are running operating systems with drastically different architectures from Windows, and we can begin to see why the risks are different and why different solutions are needed. Traditionally on Microsoft Windows systems, sandboxing has not been used, code-signing requirements have not been broadly adopted, and the OS has generally been left “open.” This has allowed malware to spread freely, causing the growth of the modern anti-virus industry.

Though the Android OS is similarly open and plagued by malware, Apple iOS implements a number of controls designed to prevent the execution of malicious or otherwise unapproved code on devices. These controls include static and dynamic code signature enforcement, which ensure that only code signed by Apple can be executed, as well as a process sandboxing model that allows for fine-grained control over which files and other resources a process can have access to. Additionally, iOS disallows execution on the stack, and supports Address Space Layout Randomization (ASLR) as well as Position Independent Executable (PIE)-compiled executables, which when combined make successful exploitation of vulnerabilities very difficult to accomplish.

Apple also requires that all apps sold via its app store be reviewed prior to distribution. This review process is not widely documented but does include at least checks related to trademark violations, copyright infringement, and the use of undocumented APIs, as well as some basic security-related checking. The OS security features combined with this review process have thus far been generally effective at keeping malware off of the iOS platform. With all of this said, the enterprise still faces significant risk from mobile apps.

**Software Procurement**

Although not exclusive to mobile, another large change that is most clearly seen in the mobile space is the complete change in software procurement (from the perspective of the enterprise). Traditionally, enterprises would procure software from large vendors, which came with some inherent level of trust. Users within the enterprise were often restricted from installing software that didn’t come via the IT procurement team. With the “Bring Your Own Device” (BYOD) movement and the consumerization of IT, this software procurement process for mobile has completely flipped. Now, enterprises in the commercial sector are embracing BYOD, finding it’s actually cheaper for the enterprise if users buy their device because users take better care of it and fewer support calls reach IT.

But as users bring their own devices, they are also bringing their own software. Instead of the software being written by a vendor and going through an IT approval process, mobile software is often being written by small development teams (including many one- or two-man shops) in jurisdictions outside the United States.

**Intentional Behaviors**

OS architectures greatly influence the attacks against the system and the attacks greatly influence the necessary defenses. On Microsoft Windows systems, we primarily see attacks involving either A) malware or B) unpatched vulnerabilities. In the case of the unpatched vulnerabilities, these are ultimately insecure programming mistakes (buffer overruns, heap corruption bugs, integer wrapping mistakes, etc.). Thus, to protect Microsoft Windows deployments, a large amount of resources needs to be spent on vulnerability mitigation via detection, patching, and API hooking in-memory. These programming errors are unintentional.

In the case of mobile apps today, the sources of risks in the software are actually intentional behaviors. That is to say, the developers added a feature or piece of functionality into the mobile app on purpose, which put the enterprise or consumer at risk.

**Examples of Intentional Behaviors in Mobile Apps**

At Appthority, we’ve examined hundreds of thousands of mobile apps for behaviors that could place an enterprise at risk. The following are a small handful of the risky intentional behaviors we’ve found in mobile apps:
Malware
The placement of malware in a mobile app is intentional. This isn’t much different than malware on the desktop, other than to say that malware evolutions and trends seem to be happening far faster on mobile than they did on the desktop.

Transmitting Contacts
Developers, especially developers of social networking apps, often transmit the contacts or address book from the device. The reason, while likely varying, generally relates to increasing the viral or network effects of the app. In other words, the developer wants to use the owner’s contacts to expand his or her customer base. There are a lot of problems with this approach. For one, whose contacts are they? For instance, if a consumer buys an iPhone and plugs it into his or her corporate desktop at work, it will likely sync with the contacts from Outlook. Those Outlook contacts are likely enterprise contacts, which is corporate data owned by the enterprise. Ignoring the fact that most apps that access the contacts on mobile devices don’t even ask for permission, if the app did ask for permission, it would be the user saying “yes,” even though the contacts would be enterprise data.

Due to the recent public outcry over an app called “Path” taking users’ contacts without permission, developers are now starting to look into hashing the contacts and then transmitting only the hashes. This is an improvement, but certainly only a start. It is important to note that transmission of the contacts is only an example; the larger issue is enterprise data sitting on a consumer-managed device with apps that have behaviors that affect the security of that data.

Location Tracking (With or Without Permission)
Everyone generally understands what location tracking is and at least some of the risks associated with carrying around a computer in your pocket that knows your location at all times. However, we’re finding that location tracking is being done a lot of different ways, by a lot of different entities. For instance, maybe an enterprise would agree to a device being location-tracked by a carrier, but a user may not make the same choice.

Location tracking also is happening in a wide variety of ways, such as via cell tower triangulation, GPS, GeoIP, etc. Some of these types of location tracking, on certain mobile platforms, require explicit permission. But does the user have the right to grant the permission if the enterprise purchases the device? We’re also seeing apps that intentionally bypass the permission model to track users without their permission. For instance, on Android, there exists a permission to track location by GeoIP. However, that permission is only required if the app leverages the API provided by Google to track location by GeoIP. We’ve seen apps that implement their own HTTP clients to leverage third-party GeoIP providers exclusively so the app developer can location track users without their permission.

In the case of mobile apps today, the sources of risks in the software are actually intentional behaviors.
Below we can see the request generated by a mobile app in order to obtain the Internet-facing IP address of the device being used. This information could be used in a way that violates corporate policy, and because of this, this app’s behavior should be subject to review prior to use in the enterprise.

```plaintext
GET /my-ip-address.php HTTP/1.1
Accept-Encoding: gzip
Host: www.ipaddresslocation.org
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)
```

**Conclusion**

The trend to move to mobile will not be reversed. Mobile, Cloud, and big data truly are changing modern computing for both enterprises and consumers. While change brings risk, modern technology allows enterprises to begin to measure that risk. Any software running on a mobile device, whether from a public app store or a private enterprise app store, needs to be inspected and validated against corporate policy, as that software can put the enterprise at risk. Even in BYOD environments, software running inside the enterprise should not be a black box.

*Anthony Bettini* is the CEO at Appthority, the leader in mobile app risk management solutions and recent winner of the most innovative company of the year award at RSA. Anthony’s current work focuses on scalable software analysis and behavioral profiling of mobile apps. His security experience comes from working for companies like Intel, McAfee, Foundstone, Bindview, and Netect. His background is in vulnerability research and he has published new vulnerabilities in Microsoft Windows, ISS Scanner, PGP, Symantec ESM, and other popular applications.
CONNECTING IDENTITY AND MOBILITY
A Secure, Scalable, and Sustainable Mobile Wallet Approach

By Siva Narendra

We are at the threshold of a strategic inflection point when it comes to associating one’s assigned identities with one’s mobile phone. There are 7 billion people on the planet and 5.2 billion of us are over the age of 15.¹ There are at least 5.6 billion active mobile phones across the globe, which means there is at least one mobile phone in use for every person over the age of 15.² With an annual mobile phone growth rate ten times higher than the population growth rate, mobile phones are about to become as universal as our wallets.

With this undeniable trend, the strategic inflection point is the migration of all identities that reside in our physical wallets to our mobile phones. Mobile wallets are already starting to revolutionize the way we get services from identity issuers such as banks and merchants (e.g., Apple’s Passbook, Google Wallet, Isis Mobile Wallet, and Tyfone’s iCashe Mobile Wallet). Mobile wallets are substantially different from digital wallets, which are commonly used in desktop computers, in large part because mobile phones are different from computers.

Like your physical wallet, mobile phones are virtually always with you, and like your computer, mobile phones are virtually always on the network. Your phone is likely more connected and available than your computer. You carry your phone with you more often than you carry your computer, and it’s possible you carry your phone more often than you carry your physical wallet.

The “always-on” network connectivity makes mobile phones ideal devices for accessing virtual world
identity services, like we do from computers today. However, if one were to really take advantage of mobile phones, it is the “always-with-you” aspect that makes this device unique. Because our phones are nearly always with us, they unlock the real potential of using a connected, intelligent device for real-world identity services. Doing so requires us to consider the security of the identity and the associated liability.

Mobile Wallet Considerations

To illustrate the importance of security, let us consider accessing commerce services in the virtual world and in the real world with our payment identities. Being able to use your always-with-you mobile phone for both virtual and real-world transactions is a reality, and this reality means the mobile wallet should be architecturally built to not trade security for convenience.

Consumers use payment identity cards with merchants in the real world and in the virtual world. The real world, of course, existed before payment identity cards were invented, and so had to fit within the expectation of merchants transacting based on physical proof of the consumer’s money, be it paper or plastic. These “what you have” factors of authentication are carried by the consumer in his or her physical wallet.

In contrast, the virtual world was created after payment identity cards were in use — it was a clean slate in terms of setting merchant-consumer interaction expectations. The virtual world, by choice, was built to not rely on physical proof. Payment proof relied purely on a “what you know” factor of authentication, such as a 16-digit card number or username and password mapped to one or more card numbers. This mapping is stored in the Cloud and it is referred to as the digital wallet. The challenge with the digital wallet is that the Cloud creates a high concentration of card numbers, producing a perfect target for thieves. These card identity numbers can be stolen remotely and cloned instantly, all through a single hacking event. Remote stealing of a high concentration of card identities from physical wallets in the real world is virtually unfeasible. This beneficial distributed security feature will have to be considered in the development of mobile wallet implementation architecture.

“Criminals will always choose the weakest link. Many digital wallets are coming to market, some from companies with no experience in physical world payments. A well-placed hack on a weak Cloud wallet could reap billions of dollars internationally,” said Nick Holland, senior analyst at Yankee Group.3

This risk of access to sensitive information stored in the Cloud is not unique to payments. It applies to all Cloud content and services, including VoIP and email.

“While Cloud services increase productivity by allowing for ubiquitous and anywhere access, it also creates centralized points of failure. There are two solutions to this problem: one option, as more services move to the centralized Cloud servers, is to migrate identity storage to be client-centric. This will help prevent a single hack from compromising millions of identities like what happened at Zappos or LinkedIn,” said Internet pioneer Dr. Steve Crocker. “The other alternative is to have completely distributed storage that syncs up between

Local Data 2011-2012 (In Billions)

<table>
<thead>
<tr>
<th>Mobile phones</th>
<th>Age of 15</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6</td>
<td>5.2</td>
<td>7</td>
</tr>
</tbody>
</table>

Source: Central Intelligence Agency, The World Factbook
various authenticated clients without any centralized storage. This will also prevent massive loss of data with a single hacking event. In either case, information and/or control access to information is distributed, making it a more robust solution.

Today, the payment industry deals with this dramatic risk difference by making the Cloud transaction fee (card not present) more expensive than the card present, real-world transaction fee — on average 70 percent more expensive. While payment services loss is a way of life, there are services such as the one provided by the security agencies of the government to its employees, where loss of identity, and therefore data, is not an option.

It is worth noting that, as per the U.S. Census Bureau in May 2012, virtual world transactions amount to less than 5 percent of total transactions. As previously mentioned, mobile phones are always on the network, and unlike your computer, mobile phones are always on you and therefore can be used for both virtual and real-world transactions. Merchants cannot tolerate an increase in fraud or transaction fees in the real world should Cloud-based, card-not-present transactions be the method of choice for mobile wallets. So, it is essential to build a solution for mobile wallets where the identity is stored in the mobile device in a distributed and secure manner for accessing services in the centralized Cloud, as previously pointed out by Dr. Crocker. If done correctly, this will prevent remote stealing of a high concentration of identities.

Secure

The most secure storage of financial identity or, for that matter, any identity is inside a smartcard chip, akin to the PIV plastic cards government employees carry. Visa and MasterCard have fostered smartcard chip technology globally and recently set a timeline for merchant adoption to be completed by 2015 in the U.S., along with merchant incentives for adoption starting in 2012.

If we were to build digital interface modules to move the smartcard chip from a “dumb” piece of plastic to a mobile phone, we would have distributed secure identity storage. This would enable secure access to not just payment services in the virtual world but also to sensitive informational services such as VoIP and email. If one were to also build area-efficient radio modules to support smartcard radio standards, then the secure identity could be used to access physical world services as well. Such architectural approaches to the mobile wallet will not only enable secure transactions for both physical and virtual worlds, but they will also provide a very convenient experience for the consumer. Users could log on and transact with no possibility of centralized points of massive identity loss in the Cloud.

Scalable

Building digital and radio modules to move smartcards from “dumb” plastic to mobile phones means we can leverage existing global standards for interfacing with...
infrastructure. This brings tremendous advantage of scale because the mobile wallet will be instantly interoperable globally, yet will become the most secure way to interact. Such scale is not accomplished by one-off merchant integration solutions like PayPal Mobile or Square Card Case, or the use of proprietary security chips such as Intel Trusted Execution Technology or the now defunct Simpay in Europe.

One more critical aspect of scale to consider is related to liability. Each of us have multiple identities — school ID card, debit cards, credit cards, loyalty cards, secure access cards, and healthcare cards, just to name a few. Yet most of us carry only one phone. Each plastic card is issued by an identity issuer, such as a government, employer, bank, or merchant. The issuer enrolls the user, determines the user’s experience, manages the relationship with the user, and is ultimately responsible for the liability of loss of the identity. Mobile wallet solutions for scalability therefore require this smartcard hardware to be neutral to device makers and mobile network operators, just like our physical wallets.

Tyfone’s iCashe Mobile Wallet along with its associated hardware, software platform, and patents is one such neutral approach. With Tyfone’s solutions, the smartcard container hardware is device and form-factor agnostic, and the software is designed for the issuer to enroll the user and manage the user’s experience.

**Sustainable**

There were 30 billion plastic cards issued in 2011 alone. If you were to line up the cards end-to-end, that amounts to nearly 1.6 million miles of plastic every year — enough to create a six-lane plastic highway between Earth and the Moon. Mobile wallets in the long run have the potential to replace billions of plastic cards that end up in landfills, while providing a secure and convenient user experience.

---

**Siva Narendra** is co-founder and CEO of Tyfone, a global innovator in mobility that provides software, security, and payment products. Siva has authored over 60 technical papers in peer reviewed conferences and journals, and has over 150 issued/pending patents. He is a frequent speaker at a wide range of forums including ones sponsored by academia and industry as well as organizations such as IEEE, FSTC, CTIA, SmartCard Alliance, and FTC. Siva serves on the executive committee of the International Solid-State Circuit Conference and has a Ph.D. in Electrical Engineering from MIT.

**REFERENCES**

TIME TO MAKE THE APPS

By Marshall Vale and Andrew Yu

The rise of the smart mobile device continues the rapid transformation of enterprise IT. Bring Your Own Device (BYOD) programs force organizations to look for new security tools to provide just the basics of secure email and calendar functions. Yet employees, managers, and executives are also using public apps on the same devices, showcasing the latest in high quality user experience and design. These constituents are now demanding the same quality and quantity of apps from their own enterprise IT organizations. Bringing the high quality user experience to internal enterprise apps will enable improvements in information access and productivity. However, organizations face a number of challenges in providing these new apps. This article will explore the strategies for producing and maintaining successful enterprise apps.

The proliferation of mobile devices forced enterprises to face several immediate security challenges. Many vendors have provided a variety of security solutions at the device level, and basic Mobile Device Management (MDM) functionality allowed for remote wipe and related device-level security capabilities. With devices secured, standard corporate information access — predominantly email, calendar, and internal browsing — was secured independently of the other applications. Yet employees found that this standard suite of applications was not enough to really take advantage of the power of these devices. What they needed were apps designed for the tasks of their day-to-day jobs.

In 2007, Andrew Yu, the Massachusetts Institute of Technology’s Mobile Platform Architect, was tasked to address the challenge of how to deliver a useful mobile experience to members of the MIT community in a BYOD environment. After evaluating existing market solutions, Yu’s team created the open MIT Mobile Framework, which quickly drew praise and popularity from other universities. In 2010, the decision was made to spin off the project into a
commercial entity called Modo Labs, Inc. Leveraging decades of experience of delivering IT solutions to enterprise, academia, and defense, the technology evolved into Kurogo, Modo Labs’ open source Mobile Optimized Middleware platform.

To see what drives the design of Kurogo, it is useful to first distill what makes a mobile app great and to consider IT application evolution. Over the past two decades, enterprise applications have moved from operating specific applications to web applications supplied to desktop browsers. These applications have been able to assume desktop resources: large screens, fast computers, and speedy networks. Providing access to every possible bit of functionality has lead to huge cluttered screens and byzantine navigation.

Mobile apps have turned those qualities on their heads. Reductions in screen size, computation power, storage, and network speeds make it difficult to transfer the same applications over to the mobile device. More significantly, the change in input methods coupled with new models of navigation have caused a major change in the structural organization of applications. Taken all together, one sees the trend of classic multi-function web applications being deconstructed. Where once there was a busy 20-function web application, there are now several apps, each with a curated selection of functionality. Rather than provide all possible photo editing tools, Instagram only provides a handful of filters. Instead of filling the screen with related travel offers for hotels and rental car deals, airline apps present the most important functionality, such as flight status and check-in. Successful apps design for the mobile context: the amalgamated context combining the physical, technical, and mental spaces that apps are used in.

Modo Labs labels two key characteristics exhibited in high quality mobile apps as glanceable and actionable. Is the user able to quickly glance at the screen and find the desired information or functionality? Screens with overly detailed information may require scrolling, an often imprecise user interaction on mobile devices. Is the information being presented actionable? Can the user make a clear decision based on the information presented? For example, an agency’s employees may often spend their time in a purchase or requisition approval application. On a mobile device, the application should present the absolute minimum information needed to make the approval decision. Detailed information should be available with inspection, but the initial information must be glanceable and actionable. After curating functionality down to basics, don’t be surprised if users continue to use their mobile devices for these purposes even when at their desktops. The apps are simply more efficient and pleasant to use.

When mobile apps are front ends to more open-ended information, such as people directories or similar hierarchies of data, how quickly the user reaches the necessary information is vitally important. We term this quality minimizing distance to the goal. How many taps does it take to navigate or find the desired information? What extra pieces of information can the software use to help layer context for the user? Yelp incorporates GPS sensor information to quickly show stores and restaurants immediately around you. You do not have to take the extra step of entering a current zip code to inform the system of your location. Incorporating geolocation or other sensor data to provide initial filtering reduces the steps needed for the users to reach their informational or functional goal.

We’ve now seen the qualities that make mobile apps different from classic desktop application approaches. Incorporating these design qualities into enterprise apps will be critical for their adoption by users and thus the success of mobile projects. The next step along this path is deciding how to build these new enterprise apps for internal consumers. There are many tools on the market today that profess to help enterprise app creation. They employ a wide variety of technologies, from closed to open source, from leveraging web technologies, to proprietary languages and runtimes. While selecting a tool can be done with conventional technical evaluations and analysts’ reports, these methods may miss out on a new trend in using a development platform as a method to foster app creation and innovation within the organization itself.

We observed earlier that a key movement of the mobile app transition is the deconstruction of applications into very small vertical slices of functionality. The people who are closest to these slices of functionality are the people who own those particular business processes in a business unit or department. Until recently, if those people needed a mobile app, they went to one of many external app shops. They would procure an app that met their particular business needs but likely would not consider proper architectural integration or sustainability. Rather than following the external
route, forward-thinking IT leaders are looking at how to enable the creation of innovative apps in a manner that incorporates architecture, security, and lifecycle. Instead of fighting the growth of apps inside their organizations, they are providing the tools to foster their creation. Ian Finley, a Gartner analyst, labels this the “citizen developer” movement, projecting that by 2014, 25 percent of all business apps will be written by these end users.1

One organization taking such an approach is Georgetown University. The university has embraced this model by not only providing a Mobile Campus app but also by opening the platform the app was built with — Modo Labs’ Kurogo — to their community. As an enabling mechanism, an IT organization can provide a platform package that already includes supported authentication methods, connections to their common organizational data, and basic business user experience elements. This greatly speeds up the creation of new secure mobile apps that gracefully fit into the enterprise’s IT architecture. Furthermore, these apps can then be shared, built upon, and re-shared, creating a positive innovation feedback loop.

Providing the right technology solution to the developers in your internal innovation community is important. A key aspect is the ability to create new apps via “mobile mashups.” A mashup application is created by aggregating third-party data, combining it into new and simple visualizations. Instead of mobilizing a specific instance of a back-end system, a mashup application can present data from multiple sources to create a more compelling user experience. Such a model fits into the key usability qualities we outlined for mobile apps earlier, **glanceable** and **actionable**. The Kurogo platform provides key tools for creating these mashups, including at the coding and visual template layers.

To ensure that community members create sustainable and secure apps, the mobile development platform should be bundled with connections to support approved authentication and authorization systems along with common data sources used within the organization, including directories, maps, and databases. By providing this platform package, the organization lays down a framework for the end user developer to create his or her own specific mobile apps, enhancing ownership of business process challenges to improve operating performance and customer satisfaction. As an example, Kurogo provides connectors for various layers of the data stack, from low level data access via XML, CSV, and SQL to higher level forms such as LDAP for people information or KML for map overlays. Organizations that have invested in Service Oriented Architecture (SOA) should benefit from their investment when creating mobile apps. Modo Labs’ experience in integrating Kurogo-based solutions for customers has shown that organizations with mature SOA environments have been able to get their apps completed faster and at a reduced cost.

With constant and rapid evolution in the mobile device space, old approaches of mandating screen resolutions or dimensions are not feasible. A developer does not have the time to test and qualify on such a variety of devices. Rather, the platforms they use must assist in handling the display of their apps. A cornerstone of the Kurogo platform is its device detection service. After categorizing the device, Kurogo is able to use the proper set of templates to deliver a tailored experience. This enables the app to meet the user experience...
characteristics laid out previously, whether the device is a basic phone or a large tablet device.

When conventional mobile apps are deployed, changing them can often be problematic. Users do not often upgrade their apps, which causes a large increase in testing requirements when upgrading data sources. Implementation of a mobile-optimized middleware to create organizational apps brings many of the conventional benefits of a multi-tier architecture to the mobile app space. Data sources can be upgraded and migrated, or implementations can be replaced without needing to push upgraded apps. Furthermore, new users can take advantage of unique mobile features such as switching from one map data source to another when the user is in different geographic locations. An employee who travels to foreign offices can quickly familiarize themselves as the app dynamically adjusts to use local informational feeds, office maps, contact information, or news alerts.

The platform should also assist in tying the new apps into the organization’s MDM strategy, including new functionality specific to managing the apps themselves. These new tools help manage the matrix of device and data ownership policies, from organization-owned or personal devices to enterprise or personal data. For example, a BYOD policy will have organizational apps with local organizational data on employee-owned phones, requiring the IT department to selectively wipe just the organization’s apps and data and not impact the operation of personal apps or data. Solutions such as Kurogo can seamlessly work with MDM solutions to provide more complete mobile app lifecycle management.

As organizations address the early challenges of the new mobile IT landscape and securing this new generation of devices, they face a question from their users: “Where are the apps?” Enterprise mobile apps commonly sit alongside highly innovative commercial apps with user-centric designs, pushing the organization to raise the bar for institutionally provided apps. The constrained resources of mobile devices have forced a deconstruction of the classic large application model, spawning many apps targeted at specific vertical functions. Providing a mobile-optimized middleware platform to a community helps small teams and individuals create new apps that target specific operational or workflow needs, allowing those closest to the process problems to innovate solutions. Fitting these new apps into MDM solutions helps secure the divide between personal and institutional devices and data. Rather than being swept over by these forces, these elements together help an organization innovate within and manage the unstoppable changing force of mobile devices.

Marshall Vale brings his experience in producing successful software platforms across multiple markets to his role as CTO of Modo Labs. At iRobot, he held the roles of Director of Software Engineering and Program Manager for the Aware robot software platform. He has also worked for MIT, Apple, and Your File System. Marshall is the co-author of a patent, with two additional patents pending. He is a graduate of the International Engineering Program at the University of Rhode Island.

Andrew Yu is an expert on smartphones and mobile devices, with more than 16 years of Internet and mobile software experience in the U.S. and Asia. In addition to his roles as CEO and Director of Modo Labs, Andrew is also the co-founder of iMobileU, a community of educational institutions with a shared interest in mobile development. He received a BA honors degree in physics and East Asian studies from Harvard University and studied electrical engineering and computer science at MIT.

REFERENCES

Securing the New Mobile Endpoint
A technology overview from IQT portfolio company Mocana

Mobile computing is here! And although it doesn’t feel much like computing (think of glowing green screens attached to desktop machines or company servers), we love it. Everyone’s drinking the proverbial Kool-Aid that always-connected mobile computing is the way to go, and in response they’re buying and upgrading iPhones, iPads, Android phones, and tablet devices for which millions of apps exist. Compared to other computing innovations, mobile computing is enjoying unprecedented growth, fueled by users’ delight — delight with mobile’s immediacy, ease of access, instant-on, mapping, and entertainment. And unlike previous devices, mobile computing crosses all population segments, from seniors to kids.

So far, so good — but what about the security elephant in the room? We in the security industry, as well as the software industry in general, recognize that the vast majority of mobile apps are not secure. Information flows to and from these apps with little to no protection. Likewise, information that is stored on a user’s mobile device is generally unsecured, leaving it open to theft or, even worse, to hijacking by viruses, worms, and so on.

Why is Mobile Security so Hard to Solve?
People often ask, “If we all know about this scary situation, why hasn’t it been addressed?” It is a technically difficult problem to solve, but is not insurmountable. Beyond the technical issues, the industry has labored under three outdated notions:

1. the device itself is the endpoint to be secured;
2. the axiom of “s×e=c” — security + experience is a constant; and
3. security is a one-size-fits-all endeavor. We believe that acceptance of these notions is largely responsible for the lack of widespread success in securing mobile computing.

Securing the Endpoint:
The App vs. The Device
To address the first point, that the device itself must be fully secured, it is important to understand that mobile devices are not dedicated, single-purpose devices, that they typically run apps developed by a variety of providers, and that a key reason for mobile’s popularity is the ease with which new apps can be installed. Therefore, traditional security methods, such as restricting communications to and from the device, are untenable.

At Mocana, we look at the problem differently. We began by assuming that all mobile devices are (or will eventually be) compromised and that non-secure and secure apps can coexist on the device. This led us to conclude that we should secure the app itself
without worrying about the device as a whole. That is, by understanding that the endpoint is the app, not the device, new security models and approaches became evident. In particular, we were able to leverage more than a million lines of proven security software code — from our NanoDefender® product, our FIPS 140-2 certified crypto libraries, and Suite B cryptographic engine — to develop Mocana Mobile App Protection™ [MAP]. MAP is an on-premise service that ruggedizes any Android or iOS binary app with a provided security posture.

Further supporting this approach, Multiple Independent Levels of Security (MILS) — a basic tenet of high assurance — implies that security must be built directly into the app. This approach is known as a self-defending app, and is the approach taken by the NSA Fishbowl project and Mocana’s Fishbowl-compliant KeyWe™ app, which we’ll discuss shortly.

**Providing Security and a Good Experience**

Just as we refuted the conventional wisdom that the device itself must be secured, we took a hard look at a second outdated notion: that security + experience is a constant. The traditional understanding has been that the stronger the security, the weaker the user experience must be. And in general, the industry has tried to convince consumers that if they want security, they must accept subpar performance in terms of speed, features, usability, and so on. The reverse is also widely believed: if you want good performance — speed, features, and usability — you must do without security. While this may be true for the average app developer dealing with severe resource limitations in order to create a 99 cent app, and who is unlikely to be an expert in either usability or security, we decided to devote the necessary resources to end the tyranny of “s+e=c”.

Mocana’s KeyWe product is a secure voice app that is compliant with the NSA Fishbowl project in which vendors develop proprietary VoIP apps to run on commercially available off-the-shelf (COTS) Android handsets. The handsets can be used to make fully-encrypted calls — secure enough so that even the most classified information can be discussed. Not only is KeyWe highly secure — it’s currently undergoing NIAP [National Information Assurance Partnership] validation — it also provides an excellent user experience thanks to Mocana’s focus on usability. Now available in app stores, KeyWe disproves the “s+e=c” axiom: it is highly secure and provides an excellent user experience.

**Understanding that Security is Not One-Size-Fits-All**

While we were working to disprove the previously discussed notions that have hindered progress in securing mobile apps, we realized that together those notions were leading people to the erroneous assumption that security is a one-size-fits-all endeavor. And that because it’s not possible to apply a giant security model suitable for a large enterprise SaaS application to a 99 cent app, you might as well give up. But we know better.

Mocana’s first products were built to secure embedded devices, which generally have severe resource limitations and extreme performance requirements. We leveraged that very same code to build MAP’s security features, such as using correct KDFs (key derivation functions) and FIPS certified cryptographic algorithms.

MAP provides all the necessary security within the technical restrictions imposed by device apps. But how can app developers actually use MAP to implement security? After all, anyone with rather rudimentary programming skills can write an app, and millions of people have done so. But the vast majority of app developers do not have the security knowledge necessary to use a security software development kit (SDK). Perhaps even worse, the added overhead of SDK calls would likely degrade the user experience.

So we took a new approach: app wrapping — the process of injecting new code into the app itself to ensure that all operations are encrypted and secured. The “new code” is tried and tested code leveraged from our NanoDefender product: an automated approach for whitelisting an application’s behavior to prevent arbitrary code execution. By providing security in this manner, we’ve successfully broken from the SDK, one-size-fits-all approach.

**Mocana’s Mobile App Protection (MAP)**

Now that we’ve explained how Mocana’s forward-thinking approach has disproven the outdated notions that hindered the development of secure apps, it’s time to look at some real technical requirements for security and how they are addressed by Mocana MAP.

At its core, MAP is an on-premise service that ruggedizes an arbitrary Android or iOS binary app with a provided security posture, which transforms the app into a self-defending app. MAP is also a console that
can be stand-alone or integrated with an app store or a Mobile Device Management (MDM) provider.

**Applicability**
Although technically any app can be ruggedized, legal requirements of iOS devices dictate that only white-label or internally developed apps may be protected. Android is a bit more flexible, requiring only the app developer’s approval before wrapping it with MAP.

**Adaptability**
If only the bad guys would quit their meddling, securing everything would be so simple. But the threat landscape is constantly changing, and so our security solutions must be able to adapt.

MAP effectively secures app endpoints by fully encrypting data at rest (whether stored on an SD card or in a database such as SimpleDB), regardless of whether device-wide encryption is available. MAP also encrypts data in transit between the app and organization via a unique VPN tunnel, again regardless of whether the device has VPN capability. For DoD devices, MAP provides two-layered protection for classified security.

**Elimination of Human Error**
Unlike an SDK approach, where the app developer would need to set security permissions in a one-size-fits-all manner, MAP automates the security integration tasks, and thereby eliminates a major source of security mishaps — the human element.

**Mocana: Your Mobile Security Solution**
As a thought leader in the security space, Mocana has made it our business to not only provide best-in-class security solutions — our customers tell us that even our competitors acknowledge our expertise, especially in app security — but to examine the security problem in general, and mobile security in particular, from the inside-out. By disproving the conventional wisdom that the device itself is the endpoint to be secured, that the axiom of “s+e=c” is unchangeable, and that security is a one-size-fits-all endeavor, we invited in new thinking, leading to the development of Mocana Mobile App Protection (MAP) and Mocana KeyWe.

*Mocana*, an IQT portfolio company, provides security products for mobile and “Internet of Things” devices. To learn more, visit www.mocana.com.
The IQT Quarterly examines trends and advances in technology. IQT has made a number of investments in information and communication technologies, and several companies in the IQT portfolio are garnering attention for their unique solutions.

**Connectify**

IQT portfolio company Connectify has gained a following among tech consumers for its software router that turns the average PC into a mobile hotspot. The software allows users to create a password-protected, ad-hoc Wi-Fi network from a single wired Internet connection. The company’s visibility has further increased since the release of Connectify Pro, which expands this capability to allow users to share 3G and 4G connections with any Wi-Fi enabled device. Connectify’s technology has been featured in a number of technology outlets including *Lifehacker* and *TechCrunch*. Connectify has been a member of the IQT portfolio since June 2011 and is headquartered in Philadelphia, PA. [www.connectify.me](http://www.connectify.me)

**Oculis Labs**

Oculis Labs, developer of computer display privacy software, has taken an innovative approach to shielding computer screens from prying eyes. The company’s flagship PrivateEye software offers protection by using a webcam to identify “shoulder surfers,” providing alerts when others are within view of a user’s screen. The company has also developed Chameleon, which obscures sensitive text while allowing authorized users to continue reading. In addition to securing a number of high-profile industry partners, Oculis has drawn the attention of media outlets like *Bloomberg News* in response to its unique products. Oculis Labs joined the IQT portfolio in July 2011 and is headquartered in Hunt Valley, MD. [www.oculislabs.com](http://www.oculislabs.com)

**WiSpry**

WiSpry, an IQT portfolio company since February 2005, develops and manufactures tunable RF components that can be integrated into wireless devices for improved signal performance. WiSpry’s small, lightweight, low-power chips enable MEMS technology, which allows mobile phones to automatically adjust to signal changes, resulting in better reception and fewer dropped calls. This tunable antenna capability is generating buzz within the mobile technology industry, and garnering endorsements from popular technology outlets such as *Gizmodo*. WiSpry is headquartered in Irvine, CA. [www.wispry.com](http://www.wispry.com)

**Mersive**

Mersive, developer of software solutions for large, multi-projector displays, was recently named by *Red Herring* as one of the top 100 technology companies in North America for 2012. The award, which has many prestigious past winners, was given in honor of Mersive’s novel technology that seamlessly stitches together imagery from multiple projectors, allowing users to create large displays of any size and shape. The company is currently working to develop new capabilities that allow mobile devices to remotely interact with Mersive-enabled displays. Mersive joined the IQT portfolio in February 2012 and is headquartered in Denver, CO. [www.mersive.com](http://www.mersive.com)