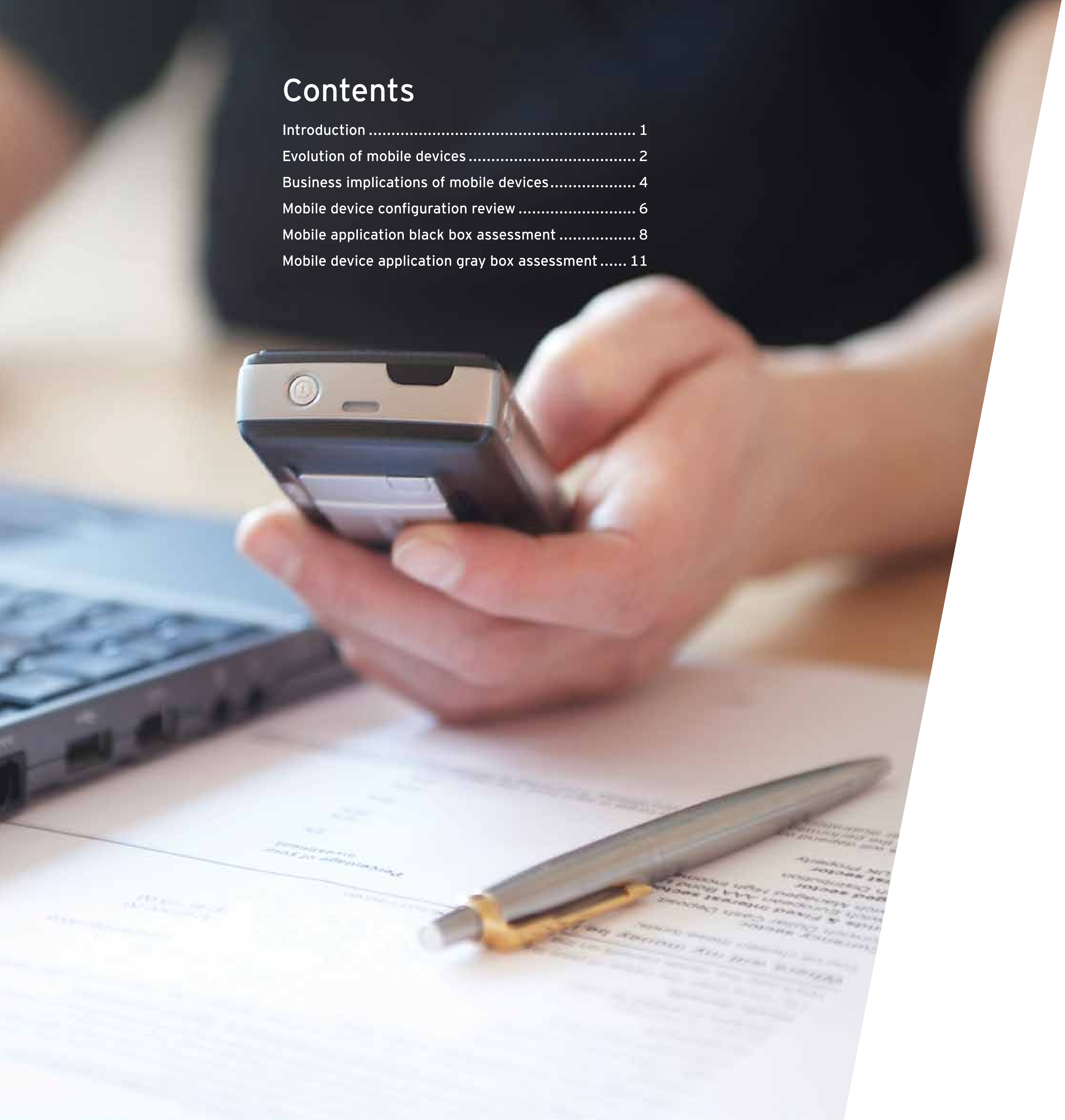
Insights on
governance, risk
and compliance

January 2012

Mobile device security

Understanding vulnerabilities and managing risks

Ernst & Young
Quality In Everything We Do



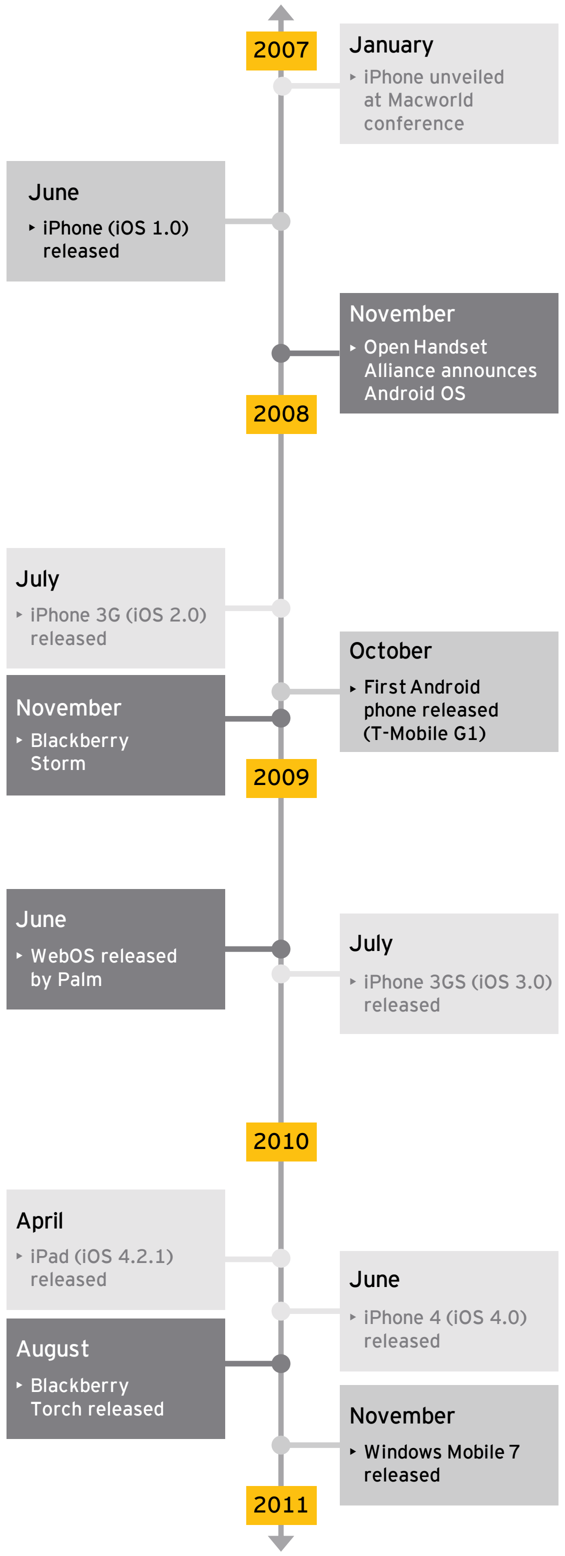
Contents

Introduction

Over the past two decades, we have witnessed significant technology advances in mobile devices, from the personal data assistants (PDAs) of the late 1990s and early 2000s to the ubiquitous and multifunctional smartphones of today. These advances have extended the virtual boundaries of the enterprise, blurring the lines between home and office and coworker and competitor by providing constant access to email, enabling new mobile business applications and allowing the access to, and storing of, sensitive company data.

In this paper, we will outline the risks related to today's most popular mobile device platforms and technologies, along with methods by which an organization may assess its exposure to these risks. Finally, we will outline means by which many of these risks may be mitigated through technical device controls, third-party software, and organizational policy. These components all contribute to an enterprise-grade mobility management program that will ultimately serve as a guide in the rapidly evolving mobile environment.

Evolution of mobile devices



2007

January
- iPhone unveiled at Macworld conference

June
- iPhone (iOS 1.0) released

November
- Open Handset Alliance announces Android OS

2008

July
- iPhone 3G (iOS 2.0) released

October
- First Android phone released (T-Mobile G1)

November
- Blackberry Storm

2009

June
- WebOS released by Palm

July
- iPhone 3GS (iOS 3.0) released

2010

April
- iPad (iOS 4.2.1) released

June
- iPhone 4 (iOS 4.0) released

August
- Blackberry Torch released

November
- Windows Mobile 7 released

2011

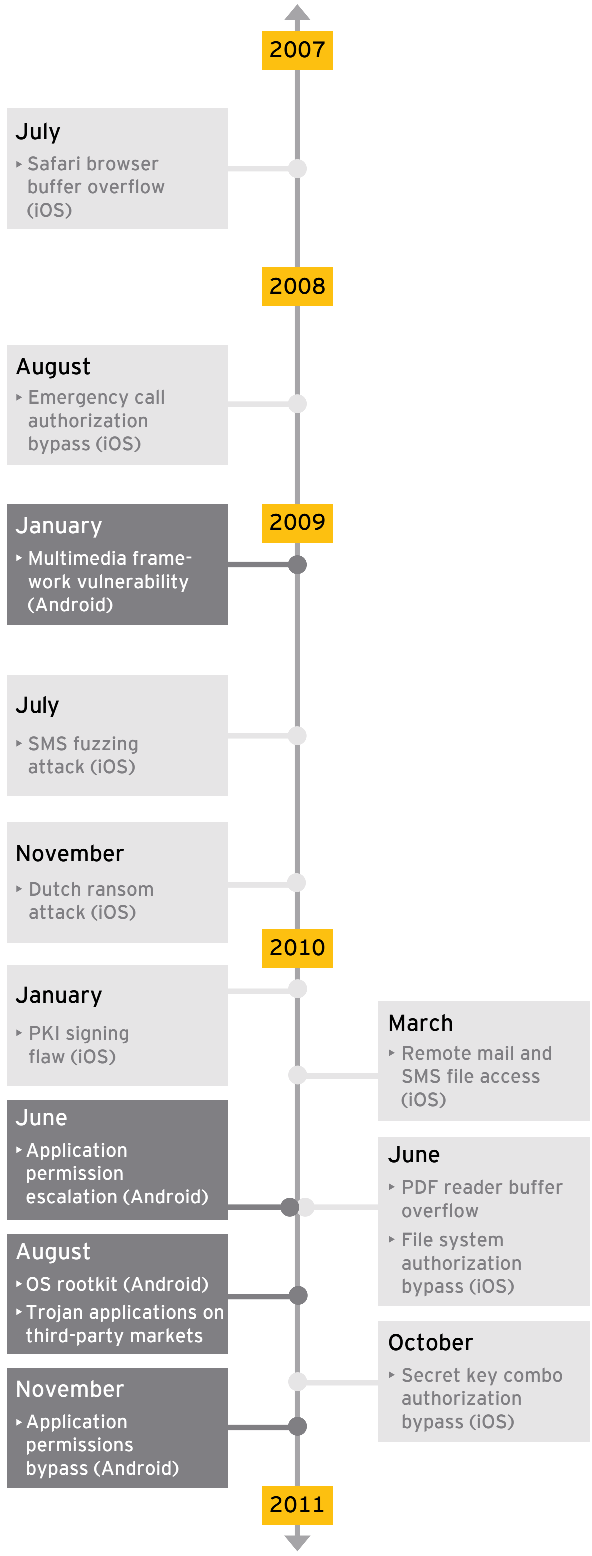
As the mobility of today's workforce continues to grow, the phrase "out of the office" is less and less relevant, and the flow of information in and out of the organization is increasing dramatically and becoming more difficult to control. The mobile workforce's demand for connectivity is driving change in the way organizations support their employees away from the office and on their personal computers. On the other side, companies are also expected to release and support robust and functional mobile device-friendly applications for their customers.

History

When the first BlackBerry smartphone was released in the early 2000s, corporations recognized the benefits of remote email and calendar access and began providing smartphones with network access to a large percentage of their workforce, effectively establishing the idea of 24-hour connectivity.

The popularity of smartphones extended beyond business users with the release of Apple's iPhone and later devices running Android, BlackBerry, Windows Mobile and Windows Phone 7 operating systems. Features expanded beyond just email and web browsing; mobile devices now have the ability to take photos, run custom applications, view rich content websites with Flash and JavaScript, connect to other devices and networks wirelessly, establish virtual private network (VPN) connections, and act as data traffic conduits for other devices (known as tethering).

Tablet PCs, such as the iPad and Galaxy, are redefining the concept of smartphones and blurring the line between mobile devices and computers. Many companies are supporting these devices as the next evolution in mobile computing.



2007

July
- Safari browser buffer overflow (iOS)

2008

August
- Emergency call authorization bypass (iOS)

January
- Multimedia framework vulnerability (Android)

2009

July
- SMS fuzzing attack (iOS)

November
- Dutch ransom attack (iOS)

2010

January
- PKI signing flaw (iOS)

March
- Remote mail and SMS file access (iOS)

June
- Application permission escalation (Android)

June
- PDF reader buffer overflow
- File system authorization bypass (iOS)

August
- OS rootkit (Android)
- Trojan applications on third-party markets

October
- Secret key combo authorization bypass (iOS)

November
- Application permissions bypass (Android)

2011

Vulnerabilities and security challenges

With the increase in mobile device capabilities and subsequent consumer adoption, these devices have become an integral part of how people accomplish tasks, both at work and in their personal lives. Although improvements in hardware and software have enabled more complex tasks to be performed on mobile devices, this functionality has also increased the attractiveness of the platform as a target for attackers. Android's "open application" model has led to multiple instances of malicious applications with hidden functionality that surreptitiously harvest user data.[1] Similarly, third-party Android application markets in China have been identified as hosting applications with administrative remote command execution capability.

Many organizations are concerned about data integrity, and increased regulation and data protection requirements have placed further obligations on organizations to properly secure data that interacts with mobile devices. As a result, higher levels of security and data protection assurance are required – potentially more than vendors or the platforms themselves are currently able to provide.

As companies around the globe look to increase the productivity of their employees or deploy new applications to appeal to an ever-increasing mobile world, corresponding security challenges present themselves. Unfortunately, the benefits and rewards of using mobile devices are sometimes counteracted by fraud and security risks.

As an example, security researchers have identified several iPhone security vulnerabilities that allowed users to bypass device restrictions and install their own firmware.[2] This may result in the users' ability to bypass many of the restrictions that prevent malicious software from running on the device. Such vulnerabilities must be considered when choosing which mobile platform(s) to support.

[1] http://googlemobile.blogspot.com/2011/03/update-on-android-market-security.html

[2] http://www.jailbreakme.com

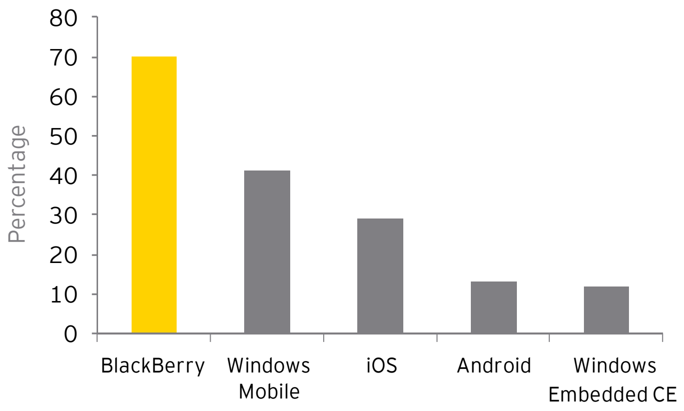
Business implications of mobile devices

Enterprise integration

With threats to mobile devices mounting, organizations need to carefully evaluate the potential risks and benefits of adopting a mobile platform strategy. Each mobile operating system design is created for a particular target audience – consumer or corporate users. Platforms designed for consumers make functional and security trade-offs to achieve simplicity and usability, while platforms designed for corporate users present less risks to an organization's environment when integrated due to the inherent secure controls embedded in the device.

Research In Motion (RIM's BlackBerry platform and supporting BlackBerry Enterprise Server environment) has historically been the front-runner in mobile device platform security, which, when combined with seamless corporate email/calendar integration, has made it an appealing platform for many organizations.

US Smartphone market share by platform

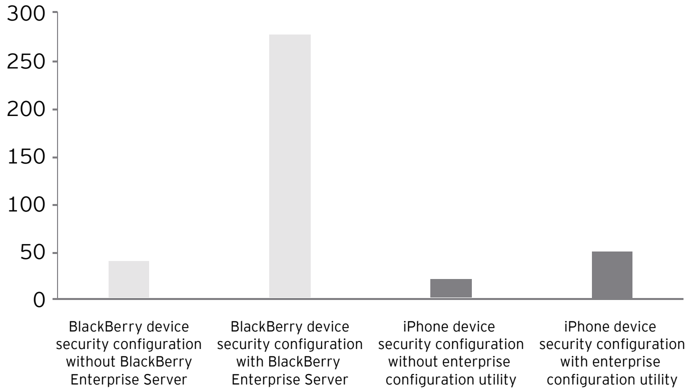


Source: comScore Reports, August 2011.

Platforms such as the iPhone and Android were designed and marketed to appeal to consumers with functional, well-designed interfaces, multimedia functionality and a customizable user experience. They were not originally intended to be secure platforms for the business world. However, as demand for business capabilities increased, these platforms began to integrate corporate functionality such as email, VPN connections and security policies. As these have been built on top of the operating systems rather than integrated at the design level, each new function may introduce new risks and require compensating controls.

As these platforms continue to mature, they will reduce the risks associated with providing enterprise support for mobile devices. With more than 250 configuration options, BlackBerry's device management infrastructure provides a granular level of security policy control for enforcement of device policy. The chart below compares the number of security settings that can be applied by an IT administrator against his organization's phones.

BlackBerry versus iPhone



BlackBerry

BlackBerry enables push technology for email, calendar, address book, instant messaging and social networking and is compatible with many enterprise mail servers, including Microsoft Exchange, IBM Lotus Domino and Novell GroupWise. When the smartphone is integrated with a BlackBerry Enterprise Server, mobile users can remotely access files, documents and other resources on the corporate intranet. Currently, the BlackBerry is the only device that can report its status after a complete remote wipe, providing assurance that corporate data has been protected.

iPhone and iPad

The original design for the iPhone did not include integrated corporate support. Enterprise email access is available via third-party tools such as Microsoft Exchange ActiveSync and standards-based services such as IMAP and LDAP. Intranet access is accomplished though VPN clients supporting Cisco IPSec, L2TP/IPSec, PPTP and SSL VPNs. Corporate profiles containing security policies and configuration information may be enabled through the iPhone Enterprise Configuration Utility.

Android

Android provides the ability to enforce password policies across devices and remotely wipe them if they are lost or compromised. Android 2.2 and 2.3 also support Microsoft Exchange calendars and auto-discovery to make it easier for users to set up and sync Microsoft Exchange accounts.

The following chart provides a comparison of corporate security features for the three most common devices supported within enterprise environments.

Feature	BlackBerry	iPhone	Android
Remote wipe capability	√	√	√
Encrypted backup files	√	√	√
Mandatory code signing	√	√	X
Type safe programming	√	◊	√
Application sandbox	√	◊	◊
Corporate policy enforcement	√	◊	◊
Full disk and memory encryption	√	◊	X
End-to-end data encryption	√	◊	X

√ Fully implemented ◊ Partially implemented X Not implemented

Application security

The rapidly expanding market of mobile devices and their open programming platforms offer corporations significant opportunities to interact with clients and customers. These devices' rich functionality supports creative innovations that are not possible through a traditional PC application. However, size and computing power limitations have forced companies to redesign their internet presence to provide mobile device users a browsing experience comparable to that of the PC. As developers redesign websites and create mobile applications, they need to consider the potential security risks and mitigate them.

Web-based mobile applications

Redesigning a website to fit the screen size of a mobile device may seem straightforward at first – simply shrink the existing site. But this approach fails to consider a mobile device's browser requirements, its support of JavaScript and embedded Flash objects, the speed of the mobile network, the computational overhead of encryption, and user input from touch-screen keyboards. Given these restrictions, developers may be inclined to choose functionality over security when trade-offs must be made.

For example, Ernst & Young has tested numerous mobile web applications where the password complexity requirements or account lockout features had been reduced or removed entirely. Restrictions on JavaScript or persistent session data have also led developers to place sensitive information and session information within the URL of every request to the server. In addition, network bandwidth limitations may encourage developers to create mobile device-formatted sites that cache additional information from web pages, potentially exposing this information if the device is compromised. During application development, developers should remember that while the screen is smaller and sensitive information may not be as readily accessible on the device, these sites are still hosted on the internet and may be accessed by traditional computers as well.

Client-based mobile applications

Apple, RIM, Microsoft, Google and other players support different operating systems and software development kits (SDKs) that developers use to create applications. Each of these platforms has a different security model that affects how developers address security within their applications. And each language has its own pitfalls and exposures that must be considered when developing an application. For instance, the iPhone programming language is based on Objective-C, where legacy modules are still vulnerable to buffer overflows. Google's guidance to individuals performing development of Android applications includes discussion of expected security do's and don'ts for both developers and users, but does not point to an official application vetting process. While Apple has an entire site dedicated to its application review process for publishing on its marketplace (http://developer.apple.com/appstore/guidelines.html), Google does not explicitly state whether or not it reviews applications before they are published on its website. Developers are asked to affirm that their application is not malicious; however, this is accomplished through a click of a button, after which the application can be published through the developer console. A lack of oversight in an application store, and the prevalence of Android applications across other sites on the internet, increases the potential of an end user accidently installing malicious software.

Mobile device configuration review

Correctly implementing a mobile device strategy across the corporate environment and mapping that strategy to local device settings can help address concerns surrounding data loss prevention, stolen devices, password policies, VPN access to intranet resources and other security issues. A mobile device configuration review can identify risks in mobile device settings and vulnerabilities in the current implementation.

Mobile device risks

The ubiquity of mobile devices in the corporate environment has allowed the further expansion of the corporate office. From a security perspective, the risks and potential effects of deploying and supporting mobile devices as a corporate tool must be understood.

Trusted clients

Mobile devices often have elevated levels of trust due to inherently strong client identification mechanisms. In the BlackBerry Enterprise Server architecture, a BlackBerry device is authenticated through a triple-DES shared-key infrastructure. This ensures that the BlackBerry unit accessing the server is a valid device (as long as the key remains uncompromised), but it does not speak to the intentions of the user. This trust also applies to other devices connected via secure channels to the environment. Due to this inherent trust of the connection, safeguards normally in place for external connections are disabled or infrequently implemented.

The iPhone application model also relies to a degree on users downloading applications from a trusted source. However, owners may bypass device restrictions through a method known as "jailbreaking." Once users jailbreak their iPhones, they can remove any policy requirements on the phone, install unapproved applications and potentially be exposed to additional security threats.

Network architecture

In a mobile device implementation, the infrastructure to control and manage mobile device connections often exists within the corporate intranet instead of a demilitarized zone (DMZ). This flat network strategy to provide mobile device data access presents the same security risks as a single-tiered wired intranet. In addition, weaknesses in vendor-advertised controls can create unexpected vulnerabilities in the security of the implementation.

Policy implementation

Compared to laptops, mobile devices often contain stronger client-side controls that can shift the security concern away from infrastructure lockdown to device lockdown. The inherent trust partly established from the tendency to trust the owners of mobile devices is fully realized when client-side controls are in place. However, an attacker can easily bypass incorrect, insufficient or weakly implemented controls, thereby leveraging the internal network's trust in the device. For example, BlackBerry devices supported by the appropriate version of BlackBerry Enterprise Server can act as modems for a laptop to access the intranet. This would bypass some device restrictions and allow a malicious user to attack the internal network from the much more functional platform of a PC.

Stolen or lost devices

A fundamental problem of mobile devices is physical access control. By their design, mobile devices are most useful outside of the office and on the move with the owner. This presents several concerns for a security administrator, as the device on the move is more likely to be lost or stolen — and subsequently used by a malicious attacker.

Considering these risks, Ernst & Young recommends assessing devices using a testing methodology specific to the risks inherent in these types of devices.

Vulnerability identification

We recommend a structured approach, consisting of both manual testing and automated reviews aimed at identifying and exploiting vulnerabilities. Specifically, we recommend assessing mobile device configurations using the following approaches:

Network accessibility

Commercial applications or custom-developed applications are used to connect to common services on the internal corporate network to test the availability of internal web servers, FTP servers, database servers and other critical infrastructure. These tests also establish connections to internal web applications that should be inaccessible to the device. The existing web browsers and proprietary web application testing tools are used to circumvent access controls.

Policy configuration

With regard to policy configuration, the local device policies determine whether end users have excessive rights or capabilities. We recommend using existing commercial tools and mobile device applications to determine the permission level provided to applications in the default corporate configuration. During a policy configuration assessment, a tester should also attempt to bypass or change policies, including those regarding device password requirements, inactivity time-out durations and installation of unapproved software.

Third-party management solutions

According to the Ernst & Young *2011 Global Information Security Survey*, 57% of respondents have made policy adjustments to mitigate the risks related to mobile computing. Many companies treat mobile devices as inherently insecure and use third-party solutions for smartphone management and increased security. These smartphone management products have enabled finer-grained tuning of security policies, segregation of work and private data spaces, and greater remote device management capabilities. However, organizations must recognize that the market is immature, with a rapidly changing vendor landscape.

Product	BlackBerry	iPhone	Android
Microsoft Exchange Server	✓	✓	✓
Novell GroupWise	✓	✓	✓
Tangoe	✓	✓	✓
Mobile Iron	✓	✓	✓
Air Watch	✓	✓	✓
Lotus Domino	✓	✓	✓
Lotus Notes Traveler	x	✓	✓
Trust Digital	x	✓	✓
Good Mail	x	✓	✓
Mobile Active Defense	x	✓	✓
Sybase Managed Mobility	✓	✓	x
BlackBerry Enterprise Server	✓	x	x

✓ Fully implemented X Not implemented

Mobile application black box assessment

One of the most significant developments in mobile device technology has been the community-driven application market. Although these small programs provide minimal functionality, they often are innovative and inexpensive, which increases their appeal. Applications are either web-based or thick clients that need to be installed on the mobile device. These two application architectures pose different risks to both the local device and the organization, and they require different front-end, or black box, testing strategies.

Web-based mobile application vulnerability identification

When assessing a web-based mobile application, we recommend that the assessor perform the testing from the perspective of an anonymous user as well as with the privileges of the various authenticated user roles in the application. Because web-based applications are accessible via the internet, the test team should use a traditional web browser on a PC along with a standard application security assessment tool set.

During assessments, scans of the web servers to identify infrastructure-level vulnerabilities are important. These scan results should then be used to identify common application issues such as those listed in the Open Web Application Security Project (OWASP) Top 10. Assessors should also perform manual techniques to fully exploit identified vulnerabilities and test for business logic and authorization flaws that automated tools often miss.

Additional web application techniques should also be applied:

Scan for vulnerabilities with proprietary and commercial tools

A non-intrusive analysis of the website should be performed, including checking content by mirroring the entire site and then checking for client-side code vulnerabilities. Using input generated from the analysis phase, proprietary tools should dynamically test the web server components for common web server and web application vulnerabilities, such as SQL injection, cross-site scripting, cross-site request forgery and directory structure. Also, execution of commercial and public domain tools should be used as deemed necessary.

Manually verify scan results

The results from the vulnerability scans should be assessed to identify probable vulnerabilities (false positive reduction). During the next phase, vulnerabilities should be validated through attempts to exploit the vulnerability and other analysis.

Web-based mobile vulnerability exploitation

Once vulnerabilities have been identified, tests should be performed to exploit them. Additional exploits attempted depend on the previous steps and may include the following:

- Insecure cookie handling
- Form spoofing manipulation
- Circumvention of application logic
- Disclosure of sensitive information
- Authentication bypass
- URL protocol handlers
- Location-based services

Analyze risk

At the conclusion of the black box review, an evaluation of identified areas of weakness should be performed, and the findings should be rated based on the risk each poses.

Device-based mobile application test environments

Applications developed for mobile devices provide challenges to testing not present in traditional applications. For example, mobile devices have limited direct access to low-level processes and exception logs. The devices also support application interaction with GPS, cameras, Bluetooth, WAP and other technologies not present in traditional PCs. To address these challenges, Ernst & Young uses two testing methods:

Simulators

Each platform provides developers with an SDK for application development and simulators of different model phones for testing and debugging purposes. These tools can also allow a tester to analyze and test applications in a variety of configurations and devices without the restrictions of a physical device. A benefit to testing within simulators is that code does not need to be signed by a trusted party to execute within the simulator.

Physical device

Testing on a physical device provides access to a number of features not available in a simulator, such as SMS, GPS, camera and Bluetooth. However, testing is restrained because of the lack of access to the underlying OS and application signing requirements.

Device-based mobile application vulnerability identification

Depending on an application's functionality, we suggest that a tester perform testing either in a simulator or on a physical device supplied by the client, or both. During the assessment, he will ascertain the application's functionality and target any internal logic controls and external connections. Because mobile applications vary in many respects, the following steps should be used as necessary for each application:

Identify application permissions

On devices such as Android and BlackBerry, applications must be granted specific access to interact with objects such as the phonebook, SMS, camera or GPS. Identifying these specific features will help the test team create a targeted test plan.

Map application functionality

Applications should be manually reviewed to identify features and functionality and identify how the application accesses different components. The assessment team should be concerned primarily with identifying external network connections, data storage, user input and permissions.

Monitor connections

Mobile devices have many means of connecting to external sources. The use of proxy tools and network sniffers to monitor every request and response should be employed and the data should be logged for later analysis. If the application uses Bluetooth or other connections, the team will pair the device to a server to capture traffic.

Review data handling

Data may reside in multiple locations throughout the use of an application. Sensitive information or application configurations may be accessible to users or unauthorized parties through various means. An assessment should identify where sensitive information is generated and analyze how the data is protected in the following areas:

- Input supplied to the application through user interaction
- Files used as input to the application
- Files created during the normal use of the application
- Application log files generated by program exceptions
- Caching mechanisms by both the application and the device that may store sensitive data in unintended locations
- Data obtained from external servers via network connections

Decompile application

When applicable, applications should be decompiled to review for known dangerous methods that may leave the application vulnerable to exploits, such as buffer overflows. Although based on Java, many mobile platforms have their own compilers that are not compatible with traditional security tools. There are a few decompilers available for BlackBerry and Android that are in beta versions, and Apple supplies a decompiler with its developer tools. These tools may be able to provide insight into the application logic and allow for limited static code analysis.

Review encryption mechanisms

Data at rest and in transit should always be protected from unauthorized parties. We recommend reviewing the use of encryption for network traffic between the device and any servers, as well as whether any files are saved by the application on the device or transferred during a backup.

Device-based mobile application vulnerability exploitation

Using the information gathered during the vulnerability identification phase, attempts to exploit the identified application vulnerabilities should be performed through some of the procedures outlined below:

Authentication and session management

Due to usability restrictions, mobile applications use many new authorization techniques, such as swipe patterns, to reduce the password complexity. Application authentication mechanisms should be tested in order to bypass these controls or access another user's data. Once authenticated, the application's session management should be reviewed. By observing how the application keeps track of users, a tester can assess if it is possible to replay a session or predictably jump to another user's session.

Authorization

Application permissions on the device should be defined with specificity. These controls prevent devices from being exploited to gain further access to the device or its features. Within the context of the application, attempts to gain access to functions that a normal user would not have permissions to execute should be performed.

Input validation

By mapping out areas of input into the application and observing the output, a security assessment can determine if client-side JavaScript can be inserted and executed in the browsers of other targeted application users. This could potentially allow for the harvesting of other users' session credentials and/or application usernames and passwords.

Data storage

Many applications collect usage data regarding their users. This data may be overly invasive and could conflict with user privacy. This data should be reviewed to determine what data is collected and stored by the application and how it is accessed. A test should be performed to determine if the data is accessible to unauthorized users or third parties.

Risk analysis

At the conclusion of the mobile black box review, the findings should be evaluated in the context of the risk each poses to the organization.

Future direction

As mobile technology companies continue to innovate over the coming years, organizations using these technologies will need to continuously assess the security implications of adopting these advancements. A consistent and agile multi-perspective mobile security risk assessment methodology will enable evaluation of the risk exposure in these systems. At Ernst & Young's Advanced Security Center, we believe security assessments, as described within this paper, are an effective method of identifying vulnerabilities and understanding their impact. Together with IT Security, Risk Management, and Internal Audit groups at our clients, we contextualize these technical findings within the business to fully understand the risk to the most critical assets. It is this teaming between technical testers and business owners that we believe will continue to be the most effective method of evaluating the security of both established and emerging technologies.

Mobile device application gray box assessment

Mobile device application gray box assessments combine traditional source code reviews (white box testing) with front-end (black box) testing techniques. The application's codebase should be examined for critical areas of functionality and for symptoms of common poor coding practices. Each of these "hot spots" in the code should be linked to the live instance of the application where manual exploit techniques can verify the existence of a security vulnerability. The recommended approach also follows this process in reverse order by reviewing the application according to our black box methodology and linking identified vulnerabilities to their cause in the codebase.

Mobile code gray box assessments are designed to:

- Prioritize high-risk areas of code
- Maximize code coverage
- Identify root causes of identified vulnerabilities

The assessment approach follows the steps outlined below.

Threat modeling

Threat modeling allows the testing team to identify first those threats that have the greatest potential impact to the application. This phase should be used to prioritize specific application components or areas of code during later phases. Using the application architecture documents provided with the application, the testing team should familiarize themselves with the general architecture and usage scenarios for the application.

Gather information

Through collaboration with the mobile application development team, documents required to assist in understanding the design and functionality of the application should be obtained. Details described in these documents will provide the foundation for all of the steps in the threat modeling process.

Undertake reconnaissance and application mapping

Understanding how the mobile application is intended to function is vital to creating a model of the application to which threats can be applied. During this step, the testing team should manually crawl and explore the live instance of the application. The team should then explore both the anonymous and authenticated portions of the application while focusing on areas of the application that handle sensitive data and functionality. The architecture, configuration, processes, users and technologies are all documented in this step and leveraged in later steps.

The areas that will be flagged for targeted testing during the next phase include:

- Administrative interfaces
- Multipart forms
- Transmission of sensitive information
- Interfaces to external or third-party applications
- Use of mobile protocols, such as SMS, MMS and WAP

Every request and response during this stage should be logged for later analysis using a combination of local proxy tools and network sniffers.

Define system and trust boundaries

During the next step of the review, an assessment team should construct a visual model of the application and its processes in a series of data flow diagrams (DFDs). A DFD will identify the system boundaries and the trust boundaries that surround each of the application's components. Identifying system boundaries gives the testing team a preliminary indication of all places where data can flow in or out of the system or its components (i.e., data entry and exit points). Later on, during the code inspection phase, the testing team should verify that proper validation and encoding techniques are being performed at each system boundary. Similarly, identifying trust boundaries will pinpoint areas of code where the testing team can verify authentication and authorization.

Map threats to functionality

After all of the DFD elements are defined, they should be mapped to threats defined in the OWASP Application Security Frame (ASF) threat categorization methodology. This exercise is intended to define "hot spots" in the application so the assessment team can create a customized test plan. Each of the items in the test plan will be evaluated fully during the targeted code inspection phase.

Vulnerability identification

The application should be reviewed with a heavy emphasis on the source code of hot spots identified in the previous phase. A black box-style assessment should be performed to identify vulnerabilities at the network or host layer in addition to application vulnerabilities not readily apparent through a pure source code review. This phase of testing should employ automated scans to complement an intensive manual inspection of application components.

Code analysis and scan

Automated scanning tools analyze the entire source code to find an initial set of security issues. This phase should utilize both commercial as well as proprietary tools to scan for symptom code and common programming errors that lead to vulnerabilities. The source code analysis phase should attempt to identify vulnerabilities that affect the application at the host, server and network layers.

Manual analysis

In this step, it is recommended that an intensive manual review of the application code is performed to find security vulnerabilities that are unique to the application's architecture. We recommend using a combination of the following techniques when reviewing the code:

- **Permission analysis** – Many platforms require the application to declare which features it will attempt to access during execution. The device will then sandbox the application to those specific features. Testers can target specific attacks against these features and attempt to bypass restrictions.
- **Control flow analysis** – This technique is used to step through logical conditions in the code. Using this method will allow testers to identify common logic flaws, such as the failure to handle exceptions, and inadequate authorization restrictions.
- **Dataflow analysis** – This technique traces data from the points of input to the points of output. This method is especially suited to identifying common input validation errors such as SQL injection and cross-site scripting.

To apply these techniques, we would recommend dividing the application into its various functional components. Each component should be examined for common insecure programming practices, which can include:

- **Authentication** – weak password requirements, username enumeration, account lockout, cookie replay attacks, backdoors
- **Authorization** – privilege escalation, inadequate separation of privileges, disclosure of confidential data, data tampering
- **Session management** – session trapping or fixation, session timeout, session hijacking, inadequate session termination, session replay, man in the middle
- **Configuration management** – unauthorized access to administrative interfaces, unauthorized access to configuration files, retrieval of clear text configuration data, overly broad privileges assigned to process and service accounts
- **Input validation** – parameter tampering, buffer overflow, cross-site scripting, SQL injection, XPATH injection, command injection
- **Data protection** – hard-coded application or user credentials, network traffic sniffing, poor key generation or key management, weak encryption, use of encoding in place of encryption
- **Exception handling** – information disclosure, denial of service
- **Auditing and logging** – log forging, log file manipulation, log file destruction
- **Caching** – keystrokes, snapshots, clipboard content and files may be cached to different storage locations on a device throughout the mobile application lifecycle
- **Password vaults** – storage of passwords in clear text in database
- **Push notifications** – one-way data transmission sent from servers to the application
- **Location-based services** – attempt to disclose or spoof location data

Review code for architecture security issues

This step is especially important if the application uses a custom security mechanism or has features to mitigate known security threats. This final code review pass is used to verify the security features that are specific to the application architecture:

- **Encryption** – Because custom encryption solutions typically are not cryptographically strong, they will be reviewed to verify that they provide adequate protection to sensitive data.
- **Protocols** – Proprietary protocols for application communication will be reviewed to determine their resistance to tampering and interception.
- **Session management** – Attempts to create custom session identifiers and session management routines will be reviewed to gauge their protection from session management errors.
- **Access restrictions** – Use of custom HTTP headers or other custom protocol elements to control access will be reviewed to ascertain protection against unauthorized access.
- **Security code** – Code written specifically to address previously identified security issues will be assessed to ascertain efficacy.
- **Server architecture** – External web services and servers used to support the application will be reviewed.

Vulnerability exploitation

This phase of the assessment is a two-part process. First, a custom test plan is developed during the first three phases of the assessment to guide the in-depth analysis of the source code for common insecure programming practices. During the second stage, a focus is placed on custom security mechanisms within the application. The code is also reviewed for architecture security issues. These are the steps we follow at Ernst & Young:

Validate identified issues

Our team analyzes the results from the vulnerability scans, eliminates false positives and creates proof-of-concept examples of exploitable vulnerabilities.

Exploit functionality unique to application

A key benefit to the gray box methodology, with its access to source code and live application, is the ability to exploit vulnerabilities to their furthest potential. In this step, we attempt to exploit authentication and authorization issues that are not apparent in the live instance of the application. These vulnerabilities may lead to unintended access to functionality or data that pose a significant risk to the business. We also will exploit flaws in business logic that is intended to control how a user performs actions in the application. These flaws typically are used to defraud application users or the company.

Link exploits to source code

As vulnerabilities are verified as exploitable, we link the exploit to the specific sections of code responsible. This allows developers to quickly understand the issue and assess the amount of effort required to remediate vulnerabilities.

Analyze risk

We evaluate the vulnerabilities that were exploited and rate the findings based on the risks each poses to the company. For each finding, we also assess the potential business impact to the company if the vulnerability is exploited. This analysis is compounded if multiple vulnerabilities are leveraged to create a greater impact.

Provide customized technical recommendations

After assessing the risk of each exploited vulnerability, we provide detailed recommendations specific to the application's architecture and codebase, including sample code when applicable. Developers then can use these recommendations to mitigate or remediate vulnerabilities and reduce risks to the application. Recommendations may also provide secure coding guidance to address vulnerabilities throughout the application.

Top 10 recommendations for mobile security

1. Add mobile security to existing employee security awareness programs.
2. Create and implement an IT policy that governs usage and ensures employees' understanding.
3. Perform threat modeling to identify the risks of moving applications to a mobile platform.
4. Train application developers in secure coding practices for mobile device platforms.
5. Limit the sensitive data transferred to mobile devices, or consider view-only access.
6. Utilize Mobile Device Management software to create an encrypted password-protected sandbox for sensitive data and enforce device-side technical policies.
7. Perform technical security assessments on mobile devices and the supporting infrastructure – focus on device-side data storage.
8. Establish a program that continually evaluates new and emerging threats in mobile platforms.
9. Increase monitoring controls around mobile device connection points when feasible.
10. Assess classic threats against web-based applications and infrastructure.

Ernst & Young

Assurance | Tax | Transactions | Advisory

About Ernst & Young
Ernst & Young is a global leader in assurance, tax, transaction and advisory services. Worldwide, our 152,000 people are united by our shared values and an unwavering commitment to quality. We make a difference by helping our people, our clients and our wider communities achieve their potential.

Ernst & Young refers to the global organization of member firms of Ernst & Young Global Limited, each of which is a separate legal entity. Ernst & Young Global Limited, a UK company limited by guarantee, does not provide services to clients. For more information about our organization, please visit www.ey.com.

About Ernst & Young's Advisory Services
The relationship between risk and performance improvement is an increasingly complex and central business challenge, with business performance directly connected to the recognition and effective management of risk. Whether your focus is on business transformation or sustaining achievement, having the right advisors on your side can make all the difference. Our 20,000 advisory professionals form one of the broadest global advisory networks of any professional organization, delivering seasoned multidisciplinary teams that work with our clients to deliver a powerful and superior client experience. We use proven, integrated methodologies to help you achieve your strategic priorities and make improvements that are sustainable for the longer term. We understand that to achieve your potential as an organization you require services that respond to your specific issues, so we bring our broad sector experience and deep subject matter knowledge to bear in a proactive and objective way. Above all, we are committed to measuring the gains and identifying where the strategy is delivering the value your business needs. It's how Ernst & Young makes a difference.

EYG no. AU1070

In line with Ernst & Young's commitment to minimize its impact on the environment, this document has been printed on paper with a high recycled content.

This publication contains information in summary form and is therefore intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. Neither EYGM Limited nor any other member of the global Ernst & Young organization can accept any responsibility for loss occasioned to any person acting or refraining from action as a result of any material in this publication. On any specific matter, reference should be made to the appropriate advisor.

How Ernst & Young makes a difference

At Ernst & Young, our services focus on our individual clients' specific business needs and issues because we recognize that each is unique to that business.

IT is a key to allowing modern organizations to compete. It offers the opportunity to become closer to customers and more focused and faster in responses, and can redefine both the effectiveness and efficiency of operations. But as opportunity grows, so does risk. Effective ITRM helps you to improve the competitive advantage of your IT operations by making these operations more cost efficient and managing down the risks related to running your systems.

Our 6,000 IT risk professionals draw on extensive personal experience to give you fresh perspectives and open, objective advice – wherever you are in the world. We work with you to develop an integrated, holistic approach to your IT risk or to deal with a specific risk and information security issue. We understand that to achieve your potential you need tailored services as much as consistent methodologies. We work to give you the benefit of our broad sector experience, our deep subject-matter knowledge and the latest insights from our work worldwide. It's how Ernst & Young makes a difference.

For more information on how we can make a difference in your organization, contact your local Ernst & Young professional or a member of our team listed below.

Contact details of our leaders

Global		
Paul van Kessel	+31 88 407 1271	paul.van.kessel@nl.ey.com
Randall J Miller	+1 312 879 3536	randall.miller@ey.com
Area		
Americas		
Michael L Herrinton	+1 703 747 0935	michael.herrinton@ey.com
Bernard R Wedge	+1 404 817 5120	bernard.wedge@ey.com
EMEIA		
Jonathan Blackmore	+44 20 7951 1616	jblackmore@uk.ey.com
Manuel Giralt Herrero	+34 91 572 7479	manuel.giraltherrero@es.ey.com
Asia-Pacific		
Jenny S Chan	+86 21 2228 2602	jenny.s.chan@cn.ey.com
Rob Perry	+61 3 9288 8639	rob.perry@au.ey.com
Japan		
Yoshihiro Azuma	+81 3 3503 1100	azuma-yshhr@shinnihon.or.jp
Haruyoshi Yokokawa	+81 3 3503 2846	yokokawa-hrysh@shinnihon.or.jp