Mobile Devices: Security Issues and Implications

(If it looks like a duck and quacks like a duck, it's probably an attack vector)

Mobile Device Prevalence

Before launching into any discussion about new technologies and the presence of or increasing threat posed by that technology, it seems reasonable to ask ourselves questions such as "how prevalent is this technology/issue?" and "is it likely to get worse or better?". After all, one could worry about the threats posed by alien superbeings or asteroids to your infrastructure, but it is unlikely to be productive or useful to your users when more pressing problems are electrical storms or lost backup tapes. To that end one doesn't need to search extensively to understand the sheer number of mobile devices in use: according to a Pew Internet Project report[1], it was found that

> *… one third of American adults – 35% – own smartphones*

and that

> *… 83% of US adults have a cell phone of some kind, and that 42% of them own a smartphone .That translates into 35% of all adults.*

Furthermore, we can easily see this increasing given the higher level of use by younger generations:

> *… 58% of Americans between the ages of 25 and 34 now own a smartphone as do 49% of those ages 18-24 and 44% of those ages 35-44.*

And what are all these devices being used for?

> *Some 87% of smartphone owners access the internet or email on their handheld, including two-thirds (68%) who do so on a typical day. When asked what device they normally use to access the internet, 25% of smartphone owners say that they mostly go online using their phone, rather than with a computer*

But what about organizations and corporate environments? In late 2010, Gartner had this to say amongst its "*Top Predictions for IT Organizations and Users for 2011 and Beyond*":

> **By 2014, 90 percent of organizations will support corporate applications on personal devices.** The trend toward supporting corporate applications on employee-owned notebooks and smartphones is already under way in many organizations and will become commonplace within four years. The main driver for adoption of mobile devices will be employees — i.e., individuals who prefer to use private consumer smartphones or notebooks for business, rather than using old-style limited enterprise devices ...

[1]http://pewinternet.org/Reports/2011/Smartphones/Summary.aspx

> **By 2013, 80 percent of businesses will support a workforce using tablets.**
> The Apple iPad is the first of what promises to be a huge wave of media tablets focused largely on content consumption, and to some extent communications, rather than content creation, with fewer features and less processing power than traditional PCs and notebooks or pen-centric tablet PCs ... enterprises will have to offer appliance-level support with a limited level of network connectivity (which will likely include access to enterprise mail and calendaring) and help desk support for connectivity issues.

The implication is clear from the above: with increasing access to corporate applications and data, that data, and in particular authentication credentials, will be either permanently or temporarily stored on those mobile devices, using whatever policies are or can be enforced by the applications being deployed as well as the underlying OS and platform. It is also likely that connections and tunnels back to corporate networks, used to allow the device and user to access data and operate as if it were *directly* connected to the corporate network, will also be available on those mobile devices.

Security Issues with Mobile Devices

Common Headaches

From the point of view of security analysis and issues affecting them, mobile devices, including smartphones, are ultimately general-purpose computers, albeit in a smaller form factor and with added functionality, particularly in relation to communications methods and channels.
There are a number of interesting aspects to the platforms relevant to security - often, unfortunately, increasing the risk associated with them, or making mitigation harder. These include:

- Platform limitations: slower and less capable CPUs, power (battery-use) restrictions, screen size and input method constraints affect whether encryption can be used and to what strength, password/passphrase complexity, how security mechanisms can interact with the user, and security software support for the platform
- Device environment: it is the exception rather than the norm that mobile devices are used in locations where the user is aware of their surroundings and the intent of those around them; so called “shoulder-surfing” is much more likely in crowded mass-transit situations, and devices are more likely to be pick-pocketed, left in taxis, or otherwise misplaced, for example.
- The platform operating system’s focus: making security seamless is non-trivial, and device and OS makers are arguably more concerned with proliferating their platforms and devices by making them easier to use, cheaper, and more accessible, which is often at odds with security mechanisms and technologies. APIs and developer support for security mechanisms and hooks within the platform’s OSs are also also in their infancy
- Breadth of data accessed, used, and stored: as mentioned previously:
    - the market for smartphones is large and increasing
    - smartphone users are using devices in place of traditional desktops, laptops and workstations for access to a broad range of systems, applications, and data
    - they are increasingly used in a corporate environments and for access to restricted data and systems: whether they be company-owned, or more likely, personal devices

we consequently see an increasing number of situations where a single, personal device is used for access to personal and corporate data.

- Access to multiple communications channels and methods: there is an emphasis on connectivity, and as such we see mobile devices with connectivity via carrier (GPRS/ EDGE, 3G, LTE), bluetooth, wireless, NFC, USB, and other channels. This increases the vectors for attack, the attack surface, and complicates policies and mitigations
- User awareness: quite simply, users are unaware or unwilling to treat mobile devices as more sensitive, or susceptible to the issues raised above. In addition, the prevalence of applications focusing on transparent access to other systems and data, often at the expense of decreased security, makes the situation worse.

Reflecting these issues, and highlighting the need to address common security problems seen across most mobile devices and applications, the OWASP project has begun drafting its “Top 10 Mobile Risks”:

- Insecure or unnecessary client-side data storage
- Lack of data protection in transit
- Personal data leakage
- Failure to protect resources with strong authentication
- Failure to implement least privilege authorization policy
- Client-side injection
- Client-side DOS
- Malicious third-party code
- Client-side buffer overflow
- Failure to apply server-side controls

These issues exist *in addition to*, rather than instead of, the well-known security problems affecting the platforms we’re all familiar with like desktops, laptops, and servers: iOS uses a version of the Safari web browser, which is used on MacOS X, Android is Linux based, and all devices and users suffer from the usual phishing and social-engineering attacks as well.

We shall now discuss some issues affecting the Android and iOS platforms specifically.
Note that the issues discussed are relevant as of this writing, but may be remediated, and others found, as time goes on.

iOS-specific security issues

In addition to the common security problems outlined above, iOS has a number of specific security shortcomings and vulnerabilities that can be used by attackers to compromise both the device, data, and potentially use that access to compromise other systems. Examples of these shortcomings include:

- Incomplete ASLR (address space layout randomization) support and use: ASLR makes exploitation much more difficult, and is present on most recent desktop OSs, but was only introduced in iOS 4.3/iPhone 3GS. In addition, although iOS 4.3’s built-in binaries fully support it, other apps must be rebuilt to gain iOS ASLR protection, and as of this writing, most 3rd party-apps have not done so
- iOS’s Safari (the default web browser) javascript engine sacrifices security for speed: to support faster javascript execution, safari is able to bypass some of the mandatory

code signing and executable protections, thus allowing cases where, should an attacker exploit javascript engine bugs, the can induce arbitrary code execution whereas normally this would be prevented

- According to Da Zovi's analysis of iOS security[2], of the in-built applications, only MobileMail and automatic screenshots make full use of iOS' "Data Protection" capabilities.
- A report from the Fraunhofer Institute for Secure Information Technology[3] indicated that many passwords stored in the iOS *keychain* are accessible (unencrypted) without having the device passcode
- Passcode protection mechanisms enforced by UI, not kernel: rather than being

[2]http://trailofbits.files.wordpress.com/2011/08/apple-ios-4-security-evaluation-whitepaper.pdf
[3]http://sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords.pdf

enforced by the kernel, the Springboard application - presented to the user via the UI - is responsible for requesting the passcode, enforcing ‘wrong passcode’ backoff, and optional device wipe protection upon reaching the threshold of incorrect attempts. Should the device be *jailbroken*[4], or be induced to run a custom application, these protections do not apply. Such a *jailbreaking* hack[5] was used by Fraunhofer Institute researchers to install scripts to extract passwords from the iOS keychain

- jailbreaking prevalence and vulnerabilities: the restrictions that Apple imposes on the platform (albeit in the name of ensuring security), particularly related to what applications

[4]removing the restriction of only allowing Apple-authorized execution of code on iOS devices using custom kernels, exploiting vulnerabilities in the device boot loader/verification chain, and so forth, allowing full access to the device filesystem

[5]http://www.pcworld.com/businesscenter/article/219245/iphone_attack_reveals_passwords_in_six_minutes.html)

can be installed, are too restrictive for some users. This leads to users jailbreaking their devices, removing almost all[6] protection mechanisms and in some cases leaving those devices susceptible to severe vulnerabilities, as the iKeeB worm[7], which exploited the default password on a remotely-accessible service installed on jailbroken iPhones, demonstrated

[6]the platforms protection mechanisms could be considered an “all or nothing” situation
[7]http://mtc.sri.com/iPhone/

- Unencrypted user input (dynamicdictionary): as far back as 2008, Jonathan Zdziarski pointed out that iOS saves user input into a plaintext file (*dynamicdictionary.dat*) so as to allow customized spelling of words by the device's user. An attacker that gained access to the device filesystem would thus have access to anything the user typed in, including sensitive information, passwords, and so forth.
- No access to some subsystems: compared with, for example, the Android platform, iOS restricts the abilities of some applications absolutely - for example, it does not allow access to certain 'sensitive' subsystems. This restricts the ability for some software to inspect and enhance the security or configuration of the device.

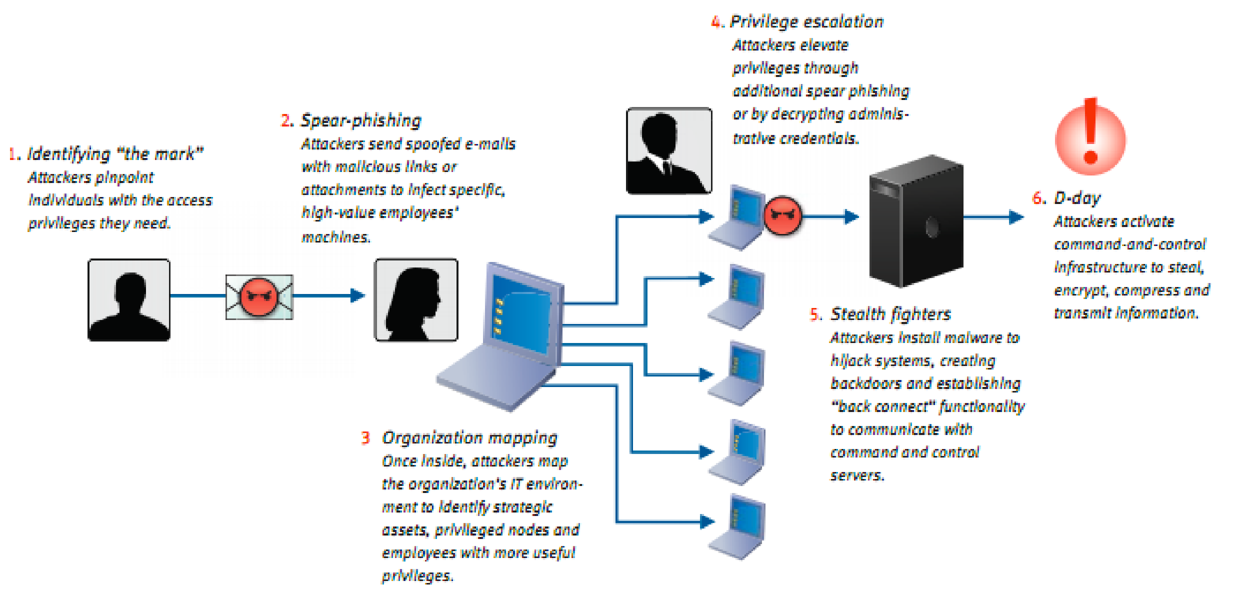
Android-specific security issues

- Android does not currently support any form of ASLR, easing the difficulty of remotely attacking and exploiting the device
- NX (eXecute Never) only recently supported: previous to Android 2.3, there was no support for non-executable memory segments, allowing attackers to deliver exploits and reliably execute them given a discovered vulnerability
- Poor encryption support: full-device encryption is only supported on android 3.0, which is not available for most devices
- Malicious apps published: a number of backdoored/malicious applications have been published to the Android app market and have been distributed to users, requiring Google to both excise the applications from the market and exercise a privledged "kill switch" to remotely remove those applications from phones. This latter capability is in itself a security risk that has yet to be fully investigated
- Preferences are stored unencrypted: application and system preferences, including in some cases authentication information, credentials, and tokens, are stored completely unencrypted on the device's filesystem. While this is not normally available without the user being prompted, an attacker with access to the device filesystem would be able to read this sensitive information
- Unsecure external SD card: the external SD memory card included with many android hardware devices can be removed and accessed completely
- Android's permission system may be confusing to non-technical users, and may desensitize users to malicious applications requesting overly-broad permissions and compromising the device or data
- Many legitimate applications request overly-broad, or too many, permissions. For example, the "Justin Beiber Wallpaper" app requests permissions to process outgoing calls, access the internet, access location information, enumerate accounts, and a number of others.

Attack Scenarios

The following diagram[8] outlines a common targeted attack scenario, which we will use as the basis for our discussion:

[8]From http://www.rsa.com/innovation/docs/11313_APT_BRF_0211.pdf


1. Identifying "the mark"
Attackers pinpoint individuals with the access privileges they need.
2. Spear-phishing
Attackers send spoofed e-mails with malicious links or attachments to infect specific, high-value employees' machines.
3 Organization mapping
Once inside, attackers map the organization's IT environment to identify strategic assets, privileged nodes and employees with more useful privileges.
4. Privilege escalation
Attackers elevate privileges through additional spear phishing or by decrypting administrative credentials.
5. Stealth fighters
Attackers install malware to hijack systems, creating backdoors and establishing "back connect" functionality to communicate with command and control servers.
6. D-day
Attackers activate command-and-control infrastructure to steal, encrypt, compress and transmit information.

Whilst this outlines an attack against a user on a laptop/desktop, the same kind of attack can be used against a smartphone user - the only difference being the vectors that can be used to deliver the initial exploit, whether it be a spoofed message, a link, or some other enticement causing the mark to activate the attack. This stage is represented in the diagram above by the progression from step one to step two, and with the introduction of smartphones, this vector can include both traditional vectors such as (inbuilt) web browsers and email clients, as well as mobile messaging systems such as SMS/MMS, mobile apps, and potentially voice[9]-based and voicemail systems, particularly if the latter features speech-to-text translation. It is not only that mobile devices contain multiple components that an attacker can use, it is also that there are greater constraints and vulnerabilities that can be exploited, as mentioned previously, and more importantly, that the level of integration between components and subsystems makes it easier for attackers to lure users into executing malicious actions unintentionally, and conversely, harder for the user to guard against such attacks.

To illustrate this problem, consider the following attack:

1. the attacker obtains the mobile phone number of the mark. This can be as simple as posing as a person the mark would be interested in conducting business with, then requesting their contact information (including cell phone number)
2. the attacker sends an MMS/SMS with a link to a malicious site. Note this would not invoke email or attachment filters as it is not sent using those mechanisms/protocols
3. the user clicks on the link in the SMS, opening the site/file in the browser
4. the malicious site/link checks the user-agent of the device and serves the appropriate

[9]we are omitting attacks against the mobile device's baseband subsystems as they require serious effort to exploit, and there are far easier methods discussed

malware to compromise the browser or other component (e.g. pdf app, video viewer, etc.). In addition the attacker also obtains device information that be used later in the attack

5. once the malware executes, the attacker executes necessary privilege-escalation exploits, completely compromising the phone
6. the attacker now has access to all authentication tokens, can establish remote tunnels, read emails or other documents, and continue with compromising internal corporate systems they are able to connect to (i.e. that the user has access to)

Should that not be sufficient, there is also a more subtle attack that can be used instead of step three: enticing the user to install the attacker's app from the platform's application store/market. While this may not look like a *better* attack, consider that:

- smartphone users are accustomed to installing new apps routinely
- it is relatively easy to publish apps to appstores/markets:
    - even for the more stringent Apple AppStore, an attacker can easily mask the malicious intent of the app, bypassing security checks required to publish it
    - given this is a targeted attack, the app is unlikely to be widely deployed or noticed by the user (assuming the attacker has a modicum of skill) and hence is unlikely to be removed by carriers or the OS maker (Apple, Google, etc.)
    - it is likely that vulnerabilities exist allowing the attacker to more easily entice the user to install applications - for example, Jon Oberheide's partial attack(s) against app installation for Android devices (http://jon.oberheide.org/files/summercon10-androidhax-jonoberheide.pdf)

An entirely different attack that can be quite successful is simply stealing the device from the user, or even gaining access to the device for a short time and installing the malicious code directly, then giving the user back the phone. Along with more esoteric attacks such as the Fraunhofer Institute researcher's "six minute" hack[10], it would be trivial to entice an inebriated user in a dark bar or nightclub to hand over their phone to an attractive member of the opposite sex who installs a malicious app, under the guise of entering contact details.

[10]http://www.businessinsider.com/iphone-password-hack-2011-2

Conclusion: How do I Protect Myself?

Perhaps the most important security lesson regarding mobile devices is: they should be treated no differently than laptops. They are, after all, complete general purpose computers, able to access, store and process a range of data. While mobile platforms have some restrictions that are security relevant, not all of the restrictions are advantages, as was extensively discussed in the section on security vulnerabilities: most of the improvements are evolutionary, or imposed due to platform limitations, rather than revolutionary improvements to security.

Given the devices are general purpose computing platforms, the most important question to ask is: *what will the device be used for, or more specifically, what kind of data will it be handling?*

If the devices are personal devices (not used for access to corporate data or systems), then simple precautions such as:

- apply updates and patches as soon as possible
- ensure encryption is used where possible, and that at least a passcode is required to unlock the device
- when using the device, be aware of who is around you and what an interested party can see (on the screen, or what you are entering)
- do not leave the device lying around, or allow people you don't trust access to it (even if it is locked)
- install a remote wiping app, and activate it as soon as you are aware the device has been lost or stolen

will reduce the risk of compromised data, accounts, and devices.

If, on the other hand, the devices are handling corporate data, or are used for accessing corporate systems, one should apply the same principles to securing the devices and addressing issues discussed, as with other corporate systems: namely, address the common problems with the particular platform, as mentioned earlier - for example, enabling encryption and device protection mechanisms, ensuring timely updates - and investigate and implement an enterprise management solution that permits restriction of apps that can be installed, policies that are enforced, and has remote wiping functionality. Finally, organizations should seriously consider user awareness training as a measure to mitigate a large number of security risks, particularly those related to social-engineering attacks mentioned earlier.