Mobile Device Management:
A Risk Discussion for IT Decision Makers

Mobile Device Management (MDM) software provides IT organizations with security-relevant capabilities that support the integration of mobile devices into enterprise networks. MDM products support enterprise-owned devices, as well as user-owned devices in what is commonly called the bring-your-own device (BYOD) scenario. This document describes the high-level architecture and capabilities of MDM solutions, and introduces key security issues to consider when deploying them.

Overview of Mobile Security

Smartphones and tablets are personal computers running a general-purpose operating system (OS), and as such their basic threat model overlaps considerably with that of conventional desktops, laptops, and servers. Yet they differ in three respects, each of which changes the threat model.

First, mobile devices are by definition portable. Naturally, this greatly increases the risk of them being lost or stolen. It also increases the risk of adversaries gaining temporary physical access, e.g. when they are left unattended in a hotel room or held for inspection by customs at a national border, and the risk that sensitive information like passwords can be observed when they are used in a public area.

Second, all smartphones and many tablets can connect to the Internet over cellular networks. This means that, even if they never leave the organization's premises, they have a path to the Internet that cannot be observed or filtered by the enterprise's network defenses, such as firewalls, proxies, and intrusion detection or prevention systems. Enterprises cannot rely on perimeter defenses alone to prevent or detect the compromise of mobile devices with cellular data connectivity.

Finally, mobile devices are generally equipped with many more physical sensors than a typical personal computer: cameras, microphones, GPS receivers, accelerometers, and so on. With the ability to collect information from the physical world, malicious software on a mobile device can compromise security even without ever connecting to the enterprise network: it can record video and audio during sensitive discussions and track the user's location in real time. A compromised mobile device could be used by an adversary to collect valuable information in less obvious ways as well. For example, public research has shown that it's feasible to derive a keystroke log from an audio recording of a user typing on a nearby keyboard.

Threat Model for Enterprise-owned Devices

Given the security-relevant characteristics described above, certain risks are involved in an enterprise deployment of mobile devices. Only enterprise-owned devices are considered here; the BYOD scenario in which personal devices are brought onto an enterprise network represents a superset of the risks discussed here and is not fully explored. However, these risks must be placed into a context that includes the realization that some of these risks exist to an even greater extent on desktop or laptop systems. The risks include:

- Malicious software or hardware may be introduced by an adversary with brief physical access to the device. This is sometimes referred to in public literature as the "evil maid" attack, since hotel housekeeping staff will have physical access to any devices left unattended in a hotel room.
- Sensitive data could be recovered from a lost or stolen device. This includes data stored on the device, and also network data that can be accessed using credentials stored on the device. If the device is powered on when the adversary acquires it, information could be recovered from RAM in addition to permanent storage.
- A remote attacker could exploit a software vulnerability to gain access to the device. Additional vulnerabilities could be exploited to execute code in the kernel, bypassing the OS's security policies.
- Unprivileged malicious applications – spyware or Trojan horses – could collect sensitive information and send it to adversaries even without exploiting any

vulnerabilities in the OS.

- When connected to an insecure network, the device's communications could be passively collected, or actively intercepted and modified by a man-in-the-middle. This can lead to the compromise of sensitive data either directly, or indirectly by making it possible to install malicious software.
- Sensitive information could be leaked inadvertently through the use of applications that are not malicious but were not designed with security in mind, e.g. "social" or advertising software that makes use of a user's location, contact list, or web history.

A well-configured MDM solution can mitigate some, but not all, of these risks. The following sections discuss the typical MDM product and its capabilities.

MDM Architecture

MDM is best understood as a complement to, rather than an augmentation of, a mobile device's security features; an MDM product's capabilities fundamentally depend on the management interfaces made available to it by the underlying mobile OS. In fact, even when an MDM vendor provides a similarly-named product for multiple mobile OS platforms, considerable differences exist between the enterprise management capabilities possible on each platform. Furthermore, neither MDM products nor any commercially-supported 3rd party software enable an enterprise to carry out arbitrary modifications to the underlying mobile OS platform. This represents a fundamental departure from the traditional managed desktop paradigm, in which 3rd party software can run highly-privileged code with power equal to that of the OS. Yet, mobile OS platforms include very powerful security-enhancing features such as application isolation and mandatory code signing that have yet to be widely implemented on desktop or server platforms. For the general-purpose use case, it is not at all clear that the lack of some enterprise controls makes these devices more vulnerable than other mobile platforms.

MDM products typically involve an agent on the mobile devices, a server component used by administrative personnel within the enterprise enclave, and an intermediary service operated by the platform vendor. The on-device agent may be a hidden part of the mobile OS itself, or it may take the form of a 3rd party app distributed through online app repositories which is permitted some additional privileges to conduct its management activities. The platform vendor's intermediary typically maintains a continuous connection to the devices to facilitate on-demand queries and commands from the enterprise.

If a product includes additional major architectural components beyond those described above, for example additional servers within the enterprise's enclave or additional intermediary servers operated by the MDM vendor, then the additional components' impact on architectural risk should be carefully considered. Key questions include:

- What happens if this component is compromised by an attacker?
- What happens if it fails and goes offline?
- Who is responsible for and has insight into this component's security and reliability, the enterprise or a third party?

When evaluating an MDM solution, decision-makers should understand its overall network architecture. Key questions include:

- How many servers will need to be deployed on the enterprise network, and what degree of access will they require to internal services such as mail, calendars, and corporate directories?
- What connections from the outside Internet will these servers need to accept?

The MDM servers should be deployed in a network's "Demilitarized Zone" (DMZ), with firewalls ensuring that they have the minimum required access to both the Internet and the enterprise intranet.

The trust relationships between the product's various components should also be well-understood before making a decision to deploy. Key questions include:

- Sensitive information should be encrypted at rest and in transit. Which components of the product have access to which cryptographic keys?
- Can the product be configured in a way that increases

convenience but may expose the cryptographic keys?

Clients and servers typically communicate with one another using protocols of some kind. Key questions include:

- Do they use strong, mutual cryptographic authentication, or can messages be easily spoofed?
- If a valid client is compromised by an adversary, what degree of access to the server can the adversary gain by exploiting these command protocols?

If the communications between a product's various components are carried over simple, industry-standard protocols, then the enterprise's technical analysts can and should independently verify vendor claims about the product's trust relationships using off-the-shelf packet inspection tools. If, on the other hand, the product relies on undocumented protocols that are proprietary to the MDM vendor, the task of independently verifying vendor claims becomes prohibitively time-consuming and expensive, as well as decreasing the product's interoperability.

Capabilities Typically Available

The following capabilities of the mobile OS are typically activated by MDM software.

Data-at-Rest Protection. Mobile device platforms offer a passcode lock feature, which is used to prevent unauthorized access to the device and also to generate a key to encrypt data stored on it, guarding against improper disclosure of confidential information if the device is lost or stolen.

The role of an MDM solution is usually only to ensure that the platform's encryption facilities are activated and used in a manner consistent with enterprise security policy, and are not disabled by individual users. As such, the capabilities of the OS should be understood first and foremost when considering mobile device encryption.

Vendors are typically eager to advertise algorithms and key size, but these are not the only factors that are important when considering mobile device encryption. Equally important are the questions of what data is encrypted and how the encryption keys are created and stored. Key questions include:

- Is the entire device encrypted, or only certain partitions?
- Are files and per-application storage areas encrypted individually?
- How are encryption keys derived from the user's passcode? How expensive would an offline brute-force attack be?
- Is there a recovery or key escrow mechanism? Can it be disabled?
- Are encryption keys cleared from RAM when no longer needed?

Whether the entire device, per-application storage areas, or separate storage partitions are encrypted is dependent upon the particular mobile OS.

In addition to managing the platform's built-in encryption features, some MDM products may independently encrypt the MDM agent application's own data as well. Be aware that although such a feature can, if implemented correctly, prevent the MDM agent's data from being recovered from a lost or stolen device, it cannot protect the MDM agent from privileged malicious software running on the device. An MDM agent application is one tenant among many on the OS, and its memory and execution can be completely controlled by an attacker who compromises a bug in the OS to gain elevated privileges.

Remote Wipe. MDM software usually also provides the ability for administrators to remotely erase a device. This is another countermeasure against the risk of sensitive data being improperly disclosed, but its effectiveness depends on the device being accessible via the network after it is lost. Key questions regarding remote wipe features include:

- How thorough is the wipe and how long does it take to complete?
- Is all data on the device erased, or only data associated with certain applications?
- Is the erased data overwritten with zeros, cryptographically made unavailable by erasing a decryption key, or just unlinked from the filesystem?
- Is the wipe command restricted to administrators, or

can users initiate a remote wipe as soon as they notice a device is missing?

- Can the MDM software be configured to automatically initiate a wipe under certain circumstances, e.g. a large number of failed authentication attempts?
- Can the MDM software verify that the wipe has occurred?

Application Policies. MDM solutions often allow administrators to create policies that govern which apps are installed on the managed devices. Depending on the features of the MDM product and underlying OS, a variety of policies can be supported. Decision makers evaluating an MDM product should understand the app installation policies the product supports and how they align with the organization's mobile security goals. Key questions include:

- To forbid the installation of unapproved apps, does the product support whitelists, or only blacklists?
- Can the product also mandate that certain required apps be installed on all managed devices?
- Does the policy identify apps by name only, or can it also permit or deny apps by other attributes like their version, developer ID or certificate, or use of security-sensitive OS services?

For example, it might be desirable for an organization to forbid the installation of any unapproved app that can track the user's location, or record audio, while still allowing apps that don't use those services.

Enterprise App Deployment. Although app deployment isn't necessarily related to MDM, some MDM products can facilitate the deployment of in-house or specially approved apps over enterprise distribution channels. Decision makers considering the use of these enterprise app distribution facilities should understand the infrastructure involved in setting up app distribution and its impact on their network attack surface. Key questions include:

- Is app deployment handled by a separate server on the enterprise network, which must be maintained and secured?
- Does this server need to accept requests directly from the Internet, or can the managed devices connect to it over a Virtual Private Network (VPN)?

It's also important to understand how much control administrators have over the cryptographic signing of apps deployed through an MDM product. Key questions include:

- Does the MDM product sign apps itself, or does it simply distribute app packages that are already signed?
- Does it verify app signatures before deployment?
- If it signs apps, does it sign them with a dedicated enterprise-provided certificate that can be revoked without impacting other MDM features?

Version Reporting. A basic feature offered by many MDMs is a capability to report which versions of the OS and applications are installed on each managed device. Most products should allow administrators to manually query version information; automation and integration can make this feature more effective. Key questions include:

- Can the product automatically send alerts to administrators or end-users when a device has fallen behind on security patches?
- Can it automatically revoke a device's credentials, or even initiate a wipe, if a device has been left in an unmanaged or unattended state for an extended period of time?
- Does the product offer a scripting interface so that enterprise administrators can create automated reports, alerts, and actions that are customized to the organization's needs?

Detection of User-initiated Platform Modification. Some MDM products can perform limited inspections of the underlying mobile OS to detect the artifacts of user-installed "jailbreak" or "root" tools. Since MDM agents typically run with limited privileges, they stand little chance of detecting sophisticated malicious software, but they can sometimes detect user misbehavior that significantly weakens the mobile device's security posture.

VPN. Mobile OSes usually have built-in support for connecting to VPN's; an MDM product can facilitate the deployment of consistent VPN configuration to all

managed devices. Decision-makers should understand the capabilities of their mobile OS, and whether a given MDM product is a good complement to them. Key questions include:

- Can the VPN be configured to connect automatically, or must the user manually initiate the connection?
- Once the device is connected to the VPN, does all traffic pass through the encrypted tunnel, or is some traffic still transmitted in cleartext?
- Can the platform prevent the user from disabling the VPN?

The mobile OS should support certificate-based VPN authentication, with individual certificates for each user. Related questions include:

- Does the MDM have features to ease the management of those certificates?
- Can it automatically revoke a user's certificate if a security violation is detected, for example a device being reported as stolen, falling too far behind on critical security patches, or having unapproved apps installed?

WiFi and Cellular Network Restrictions. In addition to using a VPN, preventing a device from connecting to insecure networks is another means of protecting sensitive network communications. Some mobile OSes may allow MDM policy to restrict which WiFi networks, or which Access Point Names for cellular networks, a device can connect to. For enterprise-controlled devices, the ability to specify a whitelist for WiFi networks remains highly desirable, despite potential incompatibility with the BYOD scenario.

Certificate Trust Management. The security of mobile devices depends to a large extent on PKI certificates, which can be used to authenticate VPN's, websites, applications, and even MDM policy itself. The OS maintains a list of trusted certificate authorities; any certificate issued by one of those CAs is accepted by the OS as legitimate. An attacker who acquires a fraudulent certificate signed by a trusted CA, or who causes the OS to trust a fraudulent CA, can intercept, decrypt, and modify all of the device's communications, and possibly install malicious software.

It's important that an enterprise understand the set of CAs its devices trust for this reason. Most MDM products allow an enterprise to add an enterprise's in-house CA to the list. Some products may also automatically add a CA associated with the MDM vendor; when evaluating a product, determine whether it does this and what the impact is on device security and trust relationships. Ideally, administrators would also be able to use an MDM solution to remove unwanted CAs and query devices' trusted CA lists.

Policy for Connection to PCs. Most mobile devices can be connected to a personal computer for data transfer, debugging, and software installation or update. A device being connected to a malicious or compromised PC can therefore lead to the compromise of the data stored on the device, or the device itself. MDM policy should be able to limit the attack surface a device exposes to connected PCs – disabling debugging, for example, or requiring that a passcode be entered for access – or limit which PCs can connect to the device. Ideally, administrators would also be able to query a log of which PCs a device has been connected to.

Conclusion

The use of a well-configured MDM solution can ensure that the built-in security features of an enterprise's devices are consistently enabled and configured to meet the organization's security needs. Organizations are encouraged to consider MDM when planning a mobile device deployment. Decision-makers should also understand, however, that MDM doesn't add new security features to a platform; it is a way to automate the configuration of the security features already provided by the platform. Therefore, there remain residual risks that MDM alone cannot mitigate; and there remain capability gaps for some high-security scenarios that MDM software cannot yet satisfy. Please refer to the related NSA Information Assurance document "Mobile Device Management: Capability Gaps for High Security Use Cases" for further discussion.

A Product of the
Network Components
and Applications Division

NSA Information Assurance
9800 Savage Rd.
Ft. Meade, MD 20755-6704
410-854-6632 DSN: 244-6632