Security Test Methodology for an Agent of a Mobile Device Management System

Keunwoo Rhee[1], Hawon Kim[1] and Hac Yun Na[1]

[1] *The Attached Institute of ETRI*
909, Jeonmin-dong, Yuseong-gu, Daejeon, 305-390, Korea
{kwrhee, wonny, hyna}@ensec.re.kr

Abstract

In this paper, we propose test items and methods for security functions of Mobile Device Management agent. Recently, many enterprises are adopting Mobile Device Management system to manage smart phones and tablet PCs used in business. However, there is no criteria to test whether such Mobile Device Management system correctly provide required security functions and whether such functions are secure. Especially, no criteria have been established as yet to test an agent which is installed on the mobile devices and directly controls the mobile devices. Therefore, we propose the first test methodology to test the Mobile Device Management agent by identifying security requirements and drawing test items and methods. The proposed items and methods are practical since we illustrate the test items, their processes, and real world test methods for Android devices.

Keywords: *Mobile Device Management agent, smart phone, tablet PC, security test*

1. Introduction

The confidential business information leakage via smart phones and tablet PCs has continued to increase. In particular, the information leakage by the loss or theft is more frequent than by hacking or malware as 'Bring Your Own Device (BYOD)' becomes a new trend. According to the 2010 annual report published by the Ponemon Institute, 88% of confidential information leakages are caused by the users' carelessness, with 35% resulting from loss or theft [1]. Therefore, enterprises are considering the adoption of Mobile Device Management (MDM) system to manage their employees' mobile devices securely.

The MDM system comprehensively manages mobile devices by monitoring their status and controlling their functions remotely. For this reason, the MDM system generally consists of server and agent on mobile device. Between these two components, the agent is more important than the server because the agent is installed on the mobile device and directly controls the device.

After installing, an MDM agent receives the instruction or the configuration and controls the mobile device according to these instruction or configuration, then reports the result of management to the MDM server.

However, there are no criteria to test these functions of an MDM agent. Therefore, we propose test items and methods to test an MDM agent. The proposed items and methods have the following significances. (1) They are the first test methodology for an MDM agent. (2) They may improve the security of an MDM agent. They can be extended to test the security of other applications on mobile devices. (3) They are practical since we describe the real world test methods for Android devices.

In order to achieve these aims, we identify security requirements in Section 2. In Section 3, we propose items and methods to test whether an MDM agent's functions which satisfy the

security requirements are securely implemented. And lastly, we present the conclusion in Section 4.

2. Security Requirements

In order to propose test items, first we identify security requirements. The security requirements provide high-level solutions against threats. Table 1 shows security requirements.

Table 1. Security Requirements

Security Requirement
R1. The MDM agent should provide secure communication channel between the MDM agent and the server.
R2. The MDM agent should communicate with the authenticated MDM server.
R3. The MDM agent should cope with disconnecting.
R4. The MDM agent should authenticate a mobile device user before accessing the mobile device and its functionality.
R5. The MDM agent should prevent modification or removal of configurations and audit data
R6. The MDM agent should control hardware modules of mobile device.
R7. The MDM agent should protect confidential data.
R8. The MDM agent should enforce minimum condition of cryptographic mechanisms.
R9. The MDM agent should control installation, removal, execution and stop the applications.
R10. The MDM agent should detect and prevent modification of operating system.

3. Security Test Methodology

In this section, we propose test items and methods for the MDM agent's security functions which satisfy the security requirements identified in Section 2. The proposed test items and methods are practical. The proposed methodology presents essential test items and their processes. In addition, the test methods can be used to test Android devices.

3.1. Test Items and Processes

Our proposed methodology contains 15 key test items. We explain these items with processes. Table 2 shows the test items and their processes. In Table 2, 'SR' means security requirements.

Evaluation results of Item13 and Item15 affect result of other evaluation items. If the MDM agent can be removed or stopped, the MDM agent cannot control the mobile device. And if the operating system is modified, the functions of the MDM agent cannot correctly work or can be bypassed.

Table 2. Test Items and Processes

Evaluation Item	SR
Item1.Encrypted communication between the MDM agent and the server. Step1.Collect the transferred packets between the MDM agent and the server. Step2.Analyze the captured packets.	R1
Item2.Management of server access profile. Step1.Find the server access profile in the mobile device. Step2.Check the profile cannot be modified. Or the MDM agent detects the modified profile.	R2

Step3.When the Step2 fails, check the MDM agent tries to access the fake server.	
Item3.Periodical check of connectivity Step1.Disconnect the communication channel between the MDM agent and the server by airplane mode, Wi-Fi off, and so on. Step2.Check the countermeasures (e.g. alerting, locking, wiping, etc.) against the disconnecting in a certain time (e.g. in 10 minutes).	R3
Item4.User authentication Step1.Set the user authentication policies (e.g. password/PIN/locking pattern, length, combination, history, expiration period, etc.). Step2.Check the MDM agent forces users to obey the authentication policies.	R4 R8
Item5.Device lock before user authentication Step1.(With Item4) Before user authentication succeeds, the MDM agent locks the interface (e.g. touch screen, USB port, external memory, Bluetooth, etc.) to access the storage or functions of the mobile device.	R4
Item6.Integrity of configuration and audit data Step1.Find the configuration and audit data in the mobile device. Step2.Check the configuration or audit data cannot be modified or deleted. Or the MDM agent detects the modification or deletion. Step3.When the Step2 fails, check the modified or deleted configuration works. And check the modified audit data are transferred to the MDM server or the deleted audit data are not transferred to the MDM server.	R5
Item7.Hardware control Step1.Set the hardware modules (e.g. camera, Bluetooth, Wi-Fi, GPS, screen capture, voice record, etc.) as enable or disable. Step2.Check a user cannot use the disabled hardware modules.	R6
Item8.Bypasssing path Step1.(With Item6 and Item7) Check a user cannot use disabled hardware modules using debugging tools, hidden methods by manufacturer, modified configuration, and so on.	R6
Item9.Encryption of confidential data Step1.Find confidential data (e.g. business data, encryption/decryption key, password, etc.) in the mobile device. Step2.Check the confidential data is encrypted.	R7
Item10.Remote locking or wiping Step1.Set the MDM server transferring ‘Lock the device’ or ‘Wipe the device’ instructions to the MDM agent. Step2.Check the MDM agent locks or wipes the mobile device immediately. When the MDM agent wipes the mobile device, the mobile device should erase all data on storage and reset the mobile device itself. Step3.Check the deleted data don’t be recovered by forensic tools.	R7
Item11.Encryption key & cryptographic data management Step1.Find stored cryptographic data such encryption/decryption keys, data which derive the keys, passwords, and so on. Step2.Check the cryptographic data are plaintext Step3.Decompile the MDM agent application.(e.g. ‘.apk’ file for Android device) Step4.Check whether there are encryption/decryption keys or data which derive the keys, passwords in the source codes.	R7
Item12.Countermeasure of authentication failure	R8

Step1.Check the counter measure (e.g. alerting, locking, wiping, etc.) when the user fails a certain number of authentication attempts.	
Item13.MDM agent removal/stop restriction Step1.Try to remove or stop the MDM agent. Step2.Check the MDM agent cannot be removed or stopped. Step3.When the Step2 fails, check the MDM agent can be reinstalled or restarted.	R9
Item14.Application installation/removal/execution/stop restriction Step1.Try to install, remove, execute, and stop applications on the mobile device. Step2.Check the MDM agent controls the application installation, removal, execution, and stop according to the configuration from the MDM server.	R9
Item15.Detection of the OS modification Step1.Try to modify the operating system on the mobile device (e.g. rooting for Android, jail breaking for iOS, etc.). Step2.Check the MDM agent detects modification of the operating system. Step3.Check the countermeasure of the modification of operating system after detection.	R10

3.2. Test Methods

In order to test the items, test tools are necessary. In this subsection, we propose the tools and test methods for Android devices although there are many tools for various operating systems and devices. The Android is the leading platform for smart phones and tablet PCs. According to the Gartner's data, Android's share of the worldwide smart phone market was 52.3% for Quarter 3 2011 [2].

Sniffing and analyzing packets: We can collect packets by mirroring or tapping. We can also capture outbound packets from the rooted device by 'Shark for root [3]'.

Exploring and editing files: In order to access and modify the system directories and files, we can use 'Root explorer [3]', 'Android Debug Bridge (ADB) [4]', 'Dalvik Debug Monitor Service (DDMS) [4]' after rooting. Sometimes, the device management software provided by manufacturer such as Samsung's KIES [5] can be used.

Executing commands: Using ADB, we can transfer files, install applications, uninstall applications, and so on. In particular, we can execute various shell commands on the rooted device.

Decompiling source codes: Generally, the source codes of application are not provided to the tester. Fortunately, we can easily decompile Android applications since the applications are developed based on Java programming language. Using 'Dex2jar [6]', we can convert an APK file to a JAR file. And we can analysis the JAR file using 'Java Decompiler [7]'.

Editing database: Android uses SQLite database to store data. We can use 'Sqlite editor [3]' to find '.db' files and edit them. If we modify the configuration of mobile device we can use restricted functions of mobile device.

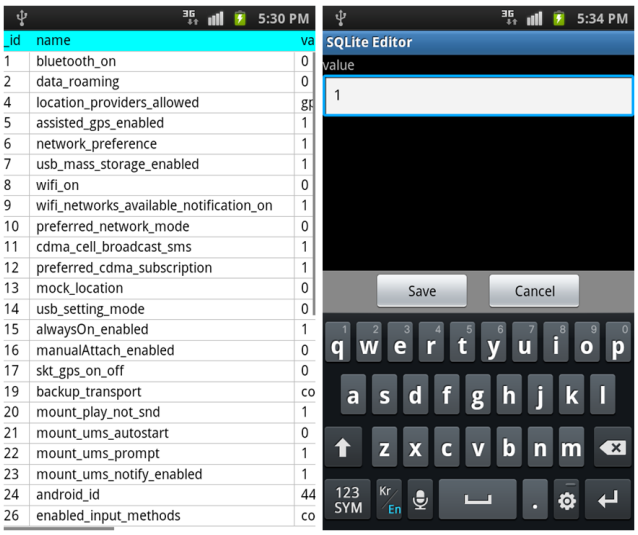

Figure 1. Modifying 'settings.db' using 'Sqlite editor'

Forensic analysis: Forensic tools are used to recover the deleted data. So, we can use forensic tools to check the device is correctly wiped.

Rooting and managing permission: In Android, the tester can get the root permission after rooting the device. By obtaining root permission, the tester can access system folders or files and execute commands which are restricted to a user. In order to use applications which need the root permission, we can use 'Superuser [3]' application.

4. Conclusion

In this paper, we analyzed threats, deduced security requirements, and proposed the first evaluation criteria and methods for the MDM agent. The proposed evaluation items and processes can be used for various platforms. In addition, they can be reused and extended to evaluate security of other applications on mobile devices. By evaluating an MDM agent according to the proposed practical criteria and methods, the security of the MDM agent will be improved. And improving security of the MDM agent will strengthen the security of the mobile devices.

Acknowledgements

Corresponding author: Hac Yun Na.

References

[1] Sybase, Inc., Why Managing Mobility Matters **(2010)**.
[2] Business Insider, GARTNER: Android Market Share Doubles, iOS Drops In Q3 **(2011)**.
[3] Google Play, https://play.google.com.
[4] Android developers, http://developer.android.com.
[5] Samsung KIES, http://www.samsung.com/sec/sppt/pcApplication.do.
[6] dex2jar, http://code.google.com/p/dex2jar.
[7] ava Decompiler, http://java.decompiler.free.fr.

Authors

Keunwoo Rhee received B.S. degree in Information and Communication Engineering and M. S. degree in Computer Engineering from Sungkyunkwan University in 2004 and 2006, respectively. He joined The Attached Institute of ETRI in 2008, and is currently a member of engineering staff of The Attached Institute of ETRI. His interests are cryptography, information security, information assurance, and security evaluation.

Hawon Kim received B.S. degrees in Computer Engineering from Pusan University in 2003 and M.S. degrees in Computer Engineering from POSTECH in 2005. He joined The Attached Institute of ETRI in 2005, and is currently a member of senior engineering staff of The Attached Institute of ETRI. His interests are security evaluation, information security, and information assurance.

Hacyun Na received B.S. and M.S. degrees in Computer Engineering from Soongsil University in 1992 and 1999, respectively. He joined The Attached Institute of ETRI in 2000, and is currently a team leader of engineering staff of The Attached Institute of ETRI. His interests are reverse engineering and security evaluation.