MobileGuardian: A Security Policy Enforcement Framework for Mobile Devices

Yong Wang

Karthik Vangury

Jason Nikolai

MobileGuardian: A Security Policy Enforcement Framework for Mobile Devices

Yong Wang
College of Business and Information System
Dakota State University
Madison, SD 57042
yong.wang@dsu.edu

Karthik Vangury, Jason Nikolai
College of Business and Information System
Dakota State University
Madison, SD 57042
{kvangury, janikolai}@pluto.dsu.edu

Abstract— **Mobile devices such as smartphones and tablets are widely used for personal and business uses. Compared to personal mobile subscribers, enterprises have more concerns about mobile device security. The challenges an enterprise may face include unlimited access to corporate resources, lack of encryption on corporate data, unwillingness to backup data, etc. Many of these issues have been resolved by auditing and enforcing security policies in enterprise networks. However, it is difficult to audit and enforce security policies on mobile devices. A substantial discrepancy exists between enterprise security policy administration and security policy enforcement. In this paper, we propose a framework, MobileGuardian, for security policy enforcement on mobile devices. Security policy enforcement is further divided into four issues, i.e., sensitive data isolation, security policy formulation, security policy testing, and security policy execution. The proposed framework is secure, flexible, and scalable. It can be adopted on any mobile platforms to implement access control, data confidentiality, security, and integrity.**

Keywords-mobile device, security policy, isolation, formulation, testing, enforcement

I. Introduction

Mobile devices such as smartphones and tablets are quickly becoming the dominant devices for accessing Internet resources. Mobile technology has changed our daily lives in many different ways, such as connecting with people, collecting information, and sharing information. According to a recent report from KPBC [1], the number of smartphone users has risen above a billion in Q3 2012 globally. Gartner estimated that 1.2 billion smartphones and tablets could be sold in 2013 [2]. It is a 46% increase compared to 821 million devices sold in 2012. As mobile devices become popular, mobile Internet grows rapidly too. Mobile traffic grows to 13% of all Internet traffic globally by November 2012 [1].

Smartphones and tablets' increasing popularity also raises many security concerns [3]. Mobile devices carry a great deal of sensitive data such as personal information, contact details, corporate data, etc. Their central data management makes them easy targets for malicious users. Since the first mobile phone virus emerged in 2004, mobile phone users have reported significant malware attacks. Malware targets mobile device valuable resources to control them and manipulate data from them. Malware attacks on the Android platform in the last seven months of 2011 increased 3,325 percent according to a report from Juniper Networks [4].

Compared to personal mobile subscribers, enterprises have more concerns about mobile device security. As companies adopt smartphones and tablets for their business, BYODs (bring your own devices) have become popular. Although BYODs let employees easily use their own devices to access corporate applications and data, it is inevitable that a mobile device includes both personal data and business data. The security of BYODs has become a new issue for enterprise administrators and IT professionals [5]. In addition, enterprises also face the challenges including unlimited access to corporate resources, lack of encryption on corporate data, and unwillingness to backup data, etc.

Many of these issues had been resolved by auditing and enforcing certain security policies on computing devices in an enterprise network. However, auditing is usually a manual, and therefore time consuming process. It is almost impossible to verify and ensure that each employee's mobile devices are in compliance with the enterprise's security policies. Further, although many companies have security policies for mobile devices used for business, many employees lack awareness of these policies and it is difficult for companies to enforce and audit these policies on employees' mobile devices too. Significant discrepancy exists between mobile device security policy administration and security policy enforcement.

In this paper, we propose a framework, MobileGuardian, for security policy enforcement on mobile devices. We divide security policy enforcement into four issues, i.e., sensitive data isolation, security policy formulation, security policy testing, and security policy execution. Unlike many proposed approach in the literature, our framework targets mobile devices in enterprise networks instead of personal mobile subscribers. The proposed framework is secure, flexible, and scalable. It can be adopted on any mobile platforms to implement access control, data confidentiality, security and integrity.

The remaining of the paper is organized as follows: Section II discusses the related work. Section III introduces the challenges of security policy enforcement on mobile devices. Section IV presents our proposed framework for security policy enforcement on mobile devices. Section V summarizes and concludes the paper.

II. Related Work

Research has been conducted on security policy enforcement on mobile devices recently. Most of the approaches proposed focus on access control and are able to monitor and report mobile apps' abnormal behaviors such as privilege escalation violation. For example, in [6], the authors present a policy enforcement framework, Apex for Android, that allows a user to selectively grant permissions to applications as well as impose constraints on the usage of resources. In [7], the authors explore the requirements and enforcement of digital rights management (DRM) policy on smartphones. DRM services ensure that protected content is accessible only by authorized phones and provider-endorsed applications. DRM services also manage access control by contextual constraints, e.g., used for a limited time, a maximum number of viewings, etc. The authors develop the Porsha system within the Android middleware to enforce DRM policies embedded in received content.

Both of the approaches in [6][7] require code changes in the corresponding mobile apps. In [8], the authors develop a solution called Aurasium that is used to repackage the application without modifying the Android OS while providing much of the security and privacy that users desire. Aurasium includes a two-step process, i.e., repackaging the application and monitoring the application's behavior. Aurasium can be used to watch an application's security and privacy violations such as attempts to retrieve a user's sensitive information, send SMS covertly to premium numbers, or access malicious IP addresses.

The approaches in [6][7][8] target personal mobile subscribers. Users select the constraints which will be used to limit access to a mobile app. However, it is hard to extend these approaches to enterprise networks. An enterprise usually has security policies already defined and expects employees to follow its policies instead of letting employees choose rules on their own. In [9], the authors propose an extension to the security architecture of the Java Virtual Machine (JVM) for mobile systems, to support fine-grained policy specification and run-time enforcement. Access control decisions are based on system state, application and system history data, as well as request specific parameters. In [10], the authors propose an approach, SecureMyDroid, in order to apply strong security policies on mobile devices by leveraging a customized release of the mobile device operating system. The approaches in [9] [10] could be extended to enterprise networks to support mobile subscribers. However, their dependency on customized OS and special version of JVM limits their usage.

III. Security Policy Enforcement Challenges on Mobile Devices

Mobile devices such as smartphones and tablets have been widely used for social networking, web surfing, calendaring, contact management, and business. Mobile device subscribers face various threats and attacks.

A. Mobile Devices Threats and Attacks

Many mobile apps in smartphones/tablets cache users' secret credentials (e.g., username and password). Mobile devices are also used for banking, business, and various other purposes. They carry a great deal of sensitive data and these data should never be disclosed to an unauthorized party. The sensitive data in mobile devices may include, but is not limited to,

- personal information such as home address, phone number, pictures
- personal contact lists
- correspondence information such as emails, text messages, MMS messages, call logs
- credit card information
- secret credentials such as passwords from different web accounts
- classified files on flash memory or memory card
- geographic location
- corporate data

All this data is located in a central place in a mobile device and it makes mobile devices easy targets for malicious users. Mobile phone virus emerged as early as in 2004. Since then, numerous malware has been reported in mobile devices. Once a mobile device is infected by malware, it is vulnerable to many threats and attacks, such as, phishing attacks, pharming attacks, vishing attacks, etc. Mobile device owners may end up with data leakage, financial loss, or invasion of privacy. Table 1 summarizes threats and attacks on mobile devices.

Mobile devices are used for both personal and business uses. Compared to personal mobile subscribers, enterprises have more concerns about mobile device security and are willing to invest more efforts to ensure their security. It is apparent that a certain standard of security requirements must be satisfied at the

Threats and Attacks		**Description**
Sniffing		Tapping or eavesdropping
Spam		Email spam and MMS message spam
Spoofing		Spoof the Caller ID or MMS Sender ID
Phishing		Steal personal information, such as user name, password, credit card account, etc
Pharming		Redirect web traffic to a malicious website followed by more specific attacks
Vishing		Voice phishing by utilizing VoIP technique
Data leakage		Unauthorized transmission of data, intentionally or unintentionally
Vulnerabilities of Webkit engine		Vulnerability allowing attackers to crash user applications and execute code
Denial of Service	Jamming	Jamming radio channel
	Flooding	MMS message flooding attacks and incoming phone call flooding attacks
	Exhausting	Battery exhaustion attack
	Blocking	Use smartphone blocking functions to disable smartphone

Table 1 Mobile Device Threats and Attacks

enterprise level and employees must ensure that their mobile devices comply with said standard. However, it is impractical for enterprise administrators and IT professionals to audit and verify the compliance of such a standard in an individual's mobile devices.

Many companies or organizations have policies regarding the use of mobile devices and accessing corporate data. However, many employees do not know the company's security policies or are not aware of the existence of such security policies. Substantial discrepancy exists on mobile devices between security policy administration and security policy enforcement. It is a challenge to enforce and audit security policies in each individual's mobile devices. However, ignoring security policy enforcement on mobile devices can have numerous negative ramifications. Confidential corporate data may fall into the wrong hands and it can cause financial loss for enterprises. The discrepancy between security policy administration and enforcement indicates that practical security policy enforcement solution on mobile devices is desirable. However, security policy enforcement is a very challenging issue.

B. *Security Policy Enforcement Challenges*

Security policy enforcement is usually resolved by auditing to enforce a certain security standard on computing devices in an enterprise network. This is still a practical approach for banking and financial industries. However, manual auditing or verification is not an option for mobile devices due to the ownership and huge number of the devices. Naive solutions such as customizing a specific application to enforce certain security policies are also insufficient because of the diversity of smartphones and tablets. Further, security policy may change constantly and new mobile devices are frequently released to market. Any solutions for security policy enforcement on mobile devices should be both flexible and scalable.

Security policies are general statements which cannot be executed on mobile devices. Security policies must be translated to machine readable languages. Since mobile devices come with various hardware and software, security policy enforcement must separate security policy definition process and execution process. In this way, security policy could be defined without considering difference of various mobile devices. Moreover, by separating security policy definition and execution, a security policy could be re-interpreted for each specific mobile device. It is easy for enterprises to adopt new mobile devices for their business.

Security policies are usually rule-based and must be tested before released to end users. Security policies could be implemented incorrectly for various reasons, e.g., misunderstanding about the policies and programming errors. Manually testing security policies is time-consuming and it is difficult to cover all testing cases. Automatic security testing is highly desirable.

Security policy enforcement is desirable on mobile devices. However, it is a challenge for enterprise administrators and IT professionals to enforce and audit security policies on an individual's mobile devices. Practical automated security policy enforcement solution is desirable.

IV. A SECURITY POLICY ENFORCEMENT FRAMEWORKN ON MOBILE DEVICES

In this paper, we propose MobileGuaradian, a security policy enforcement framework for mobile devices. The framework targets to enterprise networks and can be adopted on any mobile platforms. As discussed, auditing and naive approaches are inadequate solutions for security policy enforcement on mobile devices. A layered architecture to separate security policy definition process and execution process is highly desirable.

A. *Mobile Device Security Policy*

Security policy in general includes all the constraints used to protect a system. Our focus here is to ensure access control, confidentiality, integrity, and authentication on corporate data on mobile devices. The data to be protected may include business contact details, emails, corporate data, etc. Security policy is usually defined and maintained by an enterprise or an organization and is expected to be followed by all employees on their personal or company-issued mobile devices.

Access control allows authorized employees and mobile apps to access corporate resources such as corporate apps and data. Corporate resources can only be accessed by authorized mobile devices and endorsed mobile apps if corresponding privileges are granted. An enterprise could also limit the number of times the data can be accessed or viewed. Further, in case an employee is removed from an organization, the employee should not be able to access corporate data even if the data was downloaded and stored on local storage unit in a mobile device. Data confidentiality, integrity, and authentication ensure the validity of data and can be ensured by cryptographic operations such as encryption, message authentication code, digital signatures, etc.

B. *MobileGuardian Overview*

Our approach for security policy enforcement on mobile devices is shown in Figure 1. Security policy enforcement is divided into four components, i.e., sensitive data isolation, security policy formulation, security policy testing, and security policy execution.

- Sensitive data isolation separates sensitive data from non-sensitive data and allows the maximum flexibility of security policy.
- Security policy formulation is targeted to convert descriptive policies to mobile device understandable instructions.
- Security policy testing tests mobile security policy rules. It ensures the validation of security policy before it is issued to mobile devices.
- Security policy execution adopts and enforces security policies on each individual mobile device.

The remaining section will discuss the details of each component.

C. *Sensitive Data Isolation*

A mobile device needs to separate sensitive data from non-sensitive data and allow security policy administrators the flexibilities to assign desired data to sensitive data. The sensitive data is stored on a mobile device as cipher text and thus the

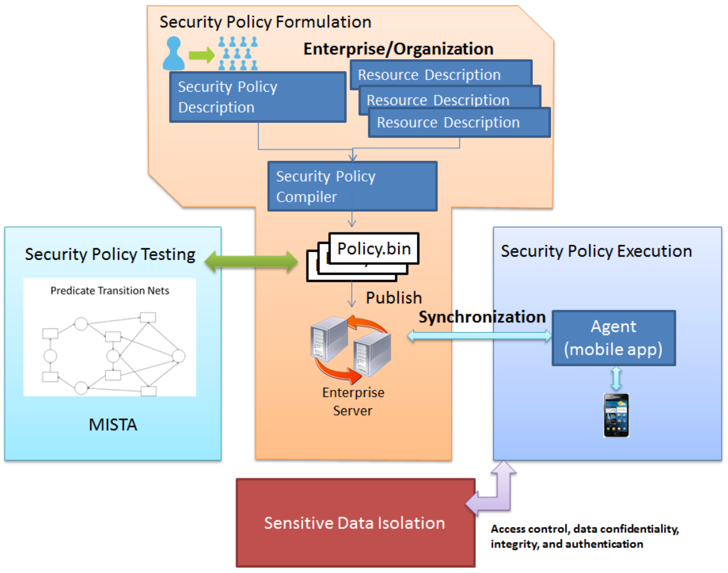

Figure 1 Mobile Device Security Policy Enforcement

information will not be disclosed if the mobile device is stolen or lost. Due to the constraints on the battery and computation power on mobile devices, ciphers need to be evaluated and benchmarked on battery consumption and encryption/decryption throughput. Allowing this capability of separation on mobile devices brings many benefits to security policy enforcement.

- Sensitive data might be an easy target for hackers. However, it is helpful to have a clear target to protect. Instead of taking extra computational power and battery to protect the entire flash or memory card, separating sensitive data from non-sensitive data is much more beneficial.
- It is easy to use security techniques, such as encryption and steganography, to protect sensitive data.
- Sensitive data isolation gives the security policy administrators the maximum capability and flexibility to customize their security policies.

The mobile device file system is divided into two areas, quarantine area (Q) and non-quarantine area (NQ):

Quarantine area: Secure area for sensitive data. Data in this area is under the protection of encryption. Ciphers and keys used for encryption are decided by key management protocols.

Non-quarantine area: Non-secure area. Data in this area is plaintext.

A security policy database (SPD) is created on both sides (in an enterprise server and on employees' mobile devices) to track services used. SPD includes the service name, IP address, port number, ciphers and the keys used. SPD is used for any packets between an enterprise network and a mobile device to ensure data can be encrypted and restored properly. An example of SPD is show in Table 2.

For example, enterprise E provides services *v* from enterprise server W: 192.168.0.1 through port p: 6666. The enterprise security policy requires that all data coming from *v* should be encrypted. With data isolation, it is much easier to define and enforce this policy. Data from services *v* will be stored in quarantine area Q. In case of theft and loss, corporate data will be safe without proper keys.

SPD Index	Service	IP	Port	Cipher	Key	Storage
1	*v*	192.168.0.1	6666	AES	…	Q
…						
…						

Table 2 Security Policy Database

Sensitive data isolation provides the following functions to support security policy enforcement:

- A quarantine area to store sensitive data
- Access control on files in the quarantine area
- Cryptographic algorithms to ensure data security such as confidentiality, integrity, and authentication
- Key distribution protocols to distribute keys to a mobile device
- Key revocation protocols to remove users' access to sensitive data

Sensitive data isolation is based on cryptographic operations. One approach is to use secret sharing algorithms. A quarantine area is encrypted using a master key. The master key is created on the fly using secrets shared by a personnel and the enterprise. In case an employee is removed from a company, it is easy for the enterprise to revoke the access rights of the employee.

D. Security Policy Formulation

Security policy formulation includes three components: security policy editor, mobile device resource editor, and security policy compiler. A security policy editor is used to define enterprise/organization security policy. A security policy description language (SPDL) is used to define security policy. A resource editor is used to describe mobile device resources. A resource description language (RDL) is used to define mobile device resources. A resource in a mobile device can be a file, a disk partition, an embedded senor, or anything which needs to be monitored. Quarantine area (Q) and non-quarantine area (NQ) are also characterized and defined as mobile device resources.

A security policy compiler applies enterprise security policy on specified resources on a mobile device. As a result, a security policy file, *policy.bin*, will be created for a certain type of mobile device. The *policy.bin* is validated then by security policy testing tool and can be published later on an enterprise server.

Assume $S = \{s_1, s_2, \dots, s_k\}$ includes all the smartphones/tablets to be supported. For each mobile device s_i, it contains certain resources r_{ij}. We use $R_i = \{r_{i1}, r_{i2}, \dots, r_{in}\}$ to describe all the resources we care about on the mobile device s_i. A security policy is usually rule based and we use $P = \{p_1, p_2, \dots, p_m\}$ to represent a policy file which include *m* rules. Let *f(p, r)* be a function which apply a rule *p* on a resource *r*. The security policy complier will combine a security policy description file with a resource description file. For example, for mobile device s_i, we have

$$f(p_1, r_{i1}), f(p_1, r_{i2}), \dots, f(p_1, r_{in})$$
$$f(p_2, r_{i1}), f(p_2, r_{i2}), \dots, f(p_2, r_{in})$$
$$\dots$$
$$f(p_m, r_{i1}), f(p_m, r_{i2}), \dots, f(p_m, r_{in})$$

These rules can be further simplified and stored in the security policy file, *policy.bin*.

Security policy is defined by a security policy administrator. A resource description file is defined for each type of mobile device only once. A security policy complier will integrate a specific security policy with a mobile device resource file. Multiple policy files could be generated if different types of mobile devices need to be supported in an enterprise.

Separating security policies and mobile device resources greatly simplifies security policy definition process and reduces the workload needed to maintain policies on each individual smartphone. First, security policy regulators can focus on general standards without worrying about actual devices. Second, the number of different mobile devices is limited and it is practical to create a resource description file for each type of mobile devices. The loose coupling between security policy definition and resource description also allows maximum flexibility to change policies or add/remove a resource easily. Third, the security policy compiler applies security policy on each resource description and converts security polices to measurable merits.

E. Security Policy Testing

Before security policy file, *policy.bin*, is published, it must be tested and validated. Security policies could be implemented incorrectly for various reasons, e.g., misunderstanding policies and programming errors. Security policy testing is another challenging issue. One approach for security policy testing is based on mutation analysis and model-based testing [11].

A mutant is a faulty rule in policy implementation. A mutant is said to be killed or detected if a failure is reported during at least one policy test execution. Mutation analysis is a widely applied method for evaluating the effectiveness of software testing techniques. It has been proven that mutation analysis is effective when used to remove injected faults from access control [11].

Based on the security policy description defined in Security Policy Formulation phase, a security policy testing model could be constructed. The security policy test model can be defined using PrT nets [12]. A PrT net consists of places (data and conditions), transitions (activities), normal and bidirectional arcs between places and transitions (input and output conditions of activities), inhibitor arcs from places to transitions (negative input conditions), and initial markings (states).

The security policy model needs to be constructed only once. A specific security policy file is only a subset of the security policy model. The security policy model can then be loaded and executable test cases are generated from the security policy model. When a security policy file, *policy.bin*, is ready to be tested, these test cases will be executed against this security policy file to find out if mutants exist.

A *policy.bin* is ready to be published after it is validated. Before releasing the *policy.bin* in the enterprise policy server, a digital signature is also generated for authentication and integration. Assume *h* is a hash function and (k_{pub}, k_{pri}) is the enterprise's public and private key. The security policy file, its corresponding smartphone identification, and their digital signature will be published in the enterprise server:

$$s_i, polic.bin, E(k_{pri}, h(s_i, polic.bin))$$

F. Security Policy Execution

Security policy will be enforced by an agent (a mobile app) running on a mobile device. The *policy.bin* can be downloaded to the subscriber's mobile devices and the security policies will be applied on the corresponding mobile resources. Mobile device security policy enforcement includes four parts:

- Mobile device security policy synchronization
- Mobile device security policy adoption
- Security policy database updating
- Security policy real time monitoring

In the synchronization process, mobile devices will synchronize with an enterprise server to check if the local security policy file is obsolete. Mobile device identification s_i will be used to retrieve the correct security policy file. The security policy file digital signature will be retrieved first and the hash value will be recovered, *h(*s_i*, policy.bin)*. If the hash value is different than the local one, the latest security policy file, *policy.bin*, will be downloaded and its signature will be further verified to ensure its integrity.

After the security policy file is validated, the *policy.bin* will be interpreted and adopted on the mobile device. The *policy.bin* is parsed first and then the security policy will be applied. At the same time when new policy file is loaded, security policy database (Table 2) is also updated to reflect the latest changes.

Security policy real time monitoring allows the agent running on mobile devices to report any policy violations to an enterprise server. The status reported may include mobile device firmware version number or operating system version number. It may also include the identifications which the mobile device fails to comply with the enterprise policies. A detailed report of employees' mobile device compliance status can be generated from the enterprise server.

G. Comparison

The proposed approach, MobileGuradian, provides a framework for security policy enforcement on mobile devices. Security policy enforcement is further divided into four issues, i.e., sensitive data isolation, security policy formulation, security policy testing, and security policy execution. The proposed framework targets security policy enforcement on mobile devices in enterprises networks and it is distinct from many existing works which target security policy enforcement for personal mobile subscribers.

The works in [6] [7] [8] propose schemes for access control on mobile devices. Their approaches target personal mobile subscribers. Users have the flexibility to select rules to watch mobile apps' abnormal behaviors. However, in an enterprise network, it is the enterprise that makes the access control polices and employees must follow the enterprise's policies. In MobileGuardian, these policies are stored in a central server in an enterprise network and are synchronized through the Internet to each individual mobile device.

The proposed framework separates security policies and mobile device resources. A compiler combines both and creates *policy.bin* for each specified mobile device. The separation of security policies and mobile device resources allow the flexibility and scalability to edit security policies and add new mobile device.

V. Conclusion

Mobile devices such as smartphones and tables are widely used for personal and business uses. A mobile device may carry a great deal of sensitive corporate data and thus it is critical for enterprise to protect mobile device security. However, huge discrepancy exists between security policy administration and security policy enforcement. In this paper, we propose a security framework, MobileGuarian, for security policy enforcement on mobile devices. MobileGuardian targets mobile devices in enterprise networks. Security policy enforcement is further divided into four components, sensitive data isolation, security policy formulation, security policy testing, and security policy execution.

The proposed framework is scalable, flexible, and secure. Our approach separates security policy definition process and enforcement process to allow maximum flexibility to define security policy and adopt new mobile devices. The proposed approach allows security policy administrators to assign corporate data in a quarantined area and put it under protection. It also provides a real time status of security policy compliance on employees' mobile devices. The framework can be extended to other computing devices, such as desktops and laptops. Our future works include continued studies on the proposed framework and implementation of the framework in Android OS.

Acknowledgements

This work is partially supported by NSF Grant No. CNS-1337529. The authors thank the anonymous reviewers for their valuable comments on this manuscript.

References

[1] M. Meeker and L. Wu, "Internet Trends," 2013.

[2] Gartner Press Release, "Gartner Says 821 Million Smart Devices Will Be Purchased Worldwide in 2012; Sales to Rise to 1.2 Billion in 2013," Barcelona, Spain, 06-Nov-2012.

[3] Y. Wang, K. Streff, and S. Raman, "Smartphone Security Challenges," *Computer (Long. Beach. Calif).*, vol. 45, no. 12, pp. 52–58, Dec. 2012.

[4] Juniper Neworks, "2011 Mobile Threats Report," 2012.

[5] Y. Wang, J. Wei, and K. Vangury, "Bring Your Own Device Security Issues and Challenges," in *The 11th Annual IEEE Consumer Communications & Networking Conference*, 2014.

[6] M. Nauman, S. Khan, and X. Zhang, "Apex : Extending Android Permission Model and Enforcement with User-defined Runtime Constraints," *Inf. Syst. J.*, no. c, pp. 328–332, 2010.

[7] M. Ongtang, K. Butler, and P. McDaniel, "Porscha: Policy oriented secure content handling in android," in *Proceedings Annual Computer Security Applications Conference ACSAC*, 2010, pp. 221–230.

[8] R. Xu, M. Park, and R. Anderson, "Aurasium : Practical Policy Enforcement for Android Applications," in *USENIX Security 12*, 2012.

[9] I. Ion, B. Dragovic, and B. Crispo, *Extending the Java Virtual Machine to Enforce Fine-Grained Security Policies in Mobile Devices*. Ieee, 2007, pp. 233–242.

[10] A. Distefano, A. Grillo, G. F. Italiano, and A. Lentini, "SecureMyDroid : Enforcing Security in the Mobile Devices Lifecycle," *Management*, pp. 1–4, 2010.

[11] D. Xu, L. Thomas, M. Kent, T. Mouelhi, and Y. Le Traon, "A model-based approach to automated testing of access control policies," in *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, 2012, pp. 209–218.

[12] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated Security Test Generation with Formal Threat Models," *Dependable Secur. Comput. IEEE Trans.*, vol. 9, no. 4, pp. 526–540, 2012.