A Resource-optimized Approach to Efficient Early Detection of Mobile Malware

Jelena Milosevic, Andreas Dittrich, Alberto Ferrante and Miroslaw Malek
Advanced Learning and Research Institute, Faculty of Informatics
Università della Svizzera italiana
Lugano, Switzerland
Email: {jelena.milosevic,andreas.dittrich,alberto.ferrante,miroslaw.malek}@usi.ch

***Abstract*—With explosive growth in the number of mobile devices mobile malware is rapidly spreading, making security one of the key issues. Existing solutions, which are mainly based on binary signatures, are not very effective.**

The main contribution of this paper is a novel methodology to design and implement secure mobile devices by offering a resource-optimized method that combines efficient, light-weight malware detection on the mobile device with high precision detection methods on cloud servers. We focus on the early detection of behavioral patterns of malware families rather than the detection of malware binary signatures. Upon detection of an attack, an alarm is raised and the damage that can be caused by the detected malware type is estimated. Furthermore, the database with behavioral patterns is continuously updated, thus keeping a device resistant to new malware families.

***Index Terms*—mobile malware; information security; distributed detection; machine learning; behavioral patterns**

I. Introduction

According to the latest Ericsson report [1], there are about 6.7 billion mobile subscriptions and by the end of 2019, they are expected to reach around 9.3 billion. However, the implications on security of such rapid deployment coupled with always-on connectivity are insufficiently understood. As stated in [2], threat alerts went up 14 percent year over year.

Mobile devices, such as smartphones and tablets, have become a prime target for attacks, since users store primarily sensitive information on them. According to McAfee Labs [3], in the last two quarters of 2013 new PC malware growth was nearly flat but appearances of new Android samples grew by 33%. While the number of mobile malware samples is increasing rapidly, the increase in the number of malware families is significantly slower [4]. Due to the increased number of malware samples, existing solutions, which are mainly focused on binary signature detection, are not very effective. Instead, behavioral malware detection, which is more concentrated on the detection of malware families, should be used.

To the best of our knowledge, current mobile malware detection methodologies are either focused on detection algorithms running on the device itself or on entirely offloading calculations to the cloud side. The advantage of malware detection on-device is that user data do not have to be sent to the network and thus, exposed to potential privacy breaches. Additionally, if the device is under attack a user receives an early notification about it and has more time to take appropriate countermeasures. However, any security mechanism targeted towards mobile systems must take their limitations into consideration as they may significantly limit the ability to run complex malware detection systems on the device. Furthermore, if a user notices that running a malware detection system drains battery quickly or slows down the operating system, chances are high that he/she will turn it off and leave the device unprotected. Due to these reasons, all computations related to detection are usually offloaded to the cloud where more sophisticated algorithms are used and detection can be done with higher confidence.

In order to trade off the advantages and drawbacks of both sides, we propose a methodology that combines a first level detection on a mobile device and a second one on a cloud infrastructure. Taking into consideration the available computational and power constraints of a mobile device, a lightweight detection algorithm is running on it. It has local knowledge about the device and with respect to that is able to detect suspicious behavior and generate an alarm upon it. Only upon generation of an alarm, data are sent to the cloud for further analysis with more complex methods. If a malware is recognised as such, a notification is sent back to the phone with possible countermeasures and a damage estimation.

The main purpose of the lightweight algorithm is to serve as a first level of protection, providing an early notification to the user about a possible malware infection. It can distinguish among different malware families with a certain confidence. If malicious behavior is observed at this level, information about the potential malware family is sent to the cloud. The primary goal of running algorithms in the cloud is to detect specific malware families with more confidence, provide an estimation about the damage that can be caused with it and advise on possible countermeasures. If computational resources and battery power of the mobile device allow it, the cloud may delegate the detection of specific malware families to the device itself. This option is introduced in the methodology since computational and power constraints of some mobile devices, for example tablets, are becoming less stringent. So we can expect that in the future, running such algorithms on these types of a devices is feasible with limited performance degradation.

In case a new malware family is observed, the system is updated, and thus more resistant to new malware. One core advantage of the methodology is that it requires updates only when new malware families appear, which is not as frequent as the appearance of new malware samples.

The rest of this paper is organized as follows. First, Section II lists contributions of the paper. In Section III the state of the art in mobile threats, behavioral malware detection, feature selection and extraction, malware detection, and distribution of malware detection algorithms is given. In section IV we present a detailed explanation of the proposed methodology. Section V compares the proposed methodology with state of the art solutions. In Section VI the impact of the methodology and its application to other areas are discussed. Finally, Section VII summarizes the paper and describes future work.

II. Problem Statement and Contribution

The proposed methodology focuses on and contributes to the following problems:

Identification of malicious behavioral patterns resembling malware and their most indicative features based on (1) the simulation of known mobile malware samples and (2) an analysis and exploitation of the semantics of the mobile device operating environment from a security point of view. Furthermore, classification of these patterns according to their representative features.

Specification of detection algorithms: (1) an efficient, lightweight algorithm with an optimized set of features for mobile malware families to be run on the mobile device and (2) a comprehensive set of powerful algorithms to detect specific behavioral malware patterns to be executed on the cloud or, in case of resource availability on the mobile device, delegated to the mobile device.

Optimization of the distribution of load, power and communication overhead among a cloud service and a set of mobile devices by taking into account the expected integrity of the mobile devices during a possible malware.

III. State of the Art

Mobile systems belong to the family of embedded systems. The embedded industry, so far, has focused more on features and other non-functional parameters, such as power consumption, but only marginally on security [5]. The rise of mobile, connected devices led to a plethora of mobile malware opportunities, exploiting known and creating new attack vectors [6], [7], [8]. The state of the art on threats, vulnerabilities and security solutions over the period 2004–2011, together with a comprehensive overview of mobile malware and predictions on future threats is surveyed in [9].

Bugs in mobile operating systems, which are far from being absent [10], are often exploited by malware. Malware is software that gains access for malicious purposes and without user's consent [11]. It includes Trojans, worms, botnets, and viruses. Additionally, users can customize smartphones by installing new applications. This creates new threats to user privacy: data can be accessed by malicious software covertly installed [12] or by exploiting security flaws through any of the available network connections. Even applications downloaded from official websites may contain hidden malicious portions of software [13].

A. Behavioral Malware Detection

Malware can be classified according to various characteristics. Among them, the usage of behavioral components as in [14] appears as a promising solution. An extensive survey covering behavioral based malware-analysis techniques and tools is given in [15]. Behavioral detection mechanisms are used in [16] to detect mobile worms, viruses and Trojans. The authors start with the extraction of key behavioral signatures. Later on, a database with malicious behavioral patterns is created and support vector machines are used in order to train a classifier with both normal and malicious data. The evaluation of both emulated and real-world malware shows that behavioral detection not only results in high detection rates but also detects unknown malware which shares certain behavioral patterns with existing patterns in the database.

B. Feature Selection and Extraction

In order to extract behavioral patterns, feature selection has to be done. An extensive survey covering the state of the art in feature selection can be found in [17], [18].

As described in [19], [9], the architecture of a generic smartphone consists of the following layers: User, application, virtual machine or guest OS, hypervisor, physical. For each functional layer – user, application and hypervisor – the authors propose distinct features that should be collected when observing a phone's behavior.

Apple, Google, and Nokia use application permissions and review (as part of market curation and signing) to protect users from malware. The effectiveness of these mechanisms against malware in a given data set was evaluated in [11]. The authors concluded that the number of permissions alone is not sufficient to identify malware. However, they could be used as part of a set of classification features, provided that all permissions common to the malware set are infrequent among non-malicious applications.

In [20], it is proposed to identify malware with sets of permissions. Their security rules classify applications based on sets of permissions rather than individual permissions to reduce the number of false positives. In [11], sending SMS messages without confirmation or accessing unique phone identifiers like the IMEI are identified as promising features for such analysis. Legitimate applications ask those permissions less often [21]. Still, using only asked permissions when identifying malware produce a high false positive rate. For example, nearly one third of applications request access to user location but far fewer request access to user location and to start at boot time. The authors state that more sophisticated rules and classification features are required in the future.

One solution for the extraction of representative features on mobile devices as input for a subsequent anomaly detection is proposed in [22]. The detection is off-loaded to a server in the network. In [23], as a feature for detecting the likelihood of malware infection, the type of applications running on a device is used. There, if a device contains an application that is present within the list of known malware applications it is labeled as infected. While observing just this feature is not enough to give a precise answer about the device being attacked, we believe that using it as one of the indicators could promises good results.

C. Attack Detection

Hidden Markov models have been successfully used to detect intrusions into regular computing systems [24]. While these techniques can be considered for the cloud service, their complexity is expected to be prohibitive for mobile devices.

Initial work has been done on intrusion detection systems for mobile devices [9]. One approach proposes the use of simplified intrusion detection systems on mobile phones [25]. A local detection of attacks by monitoring system parameters (e.g., CPU usage) and reaction at the network level is demonstrated in [26]. The detection system is based on a database of virus signatures only and it is not able to detect unknown attacks. A neural network-based system to detect improper use of mobile phones is proposed in [27]. This work is obsolete since it limits its scope just to analog mobile phones. The authors of [26] propose a framework based on host-based intrusion detection models for mobile devices. A cloud-based intrusion detection system that relies on a cloud-side copy of the mobile device has been proposed in [28]. The purpose of this system is to implement a complex attack detection system with minimized impact on mobile performance.

Several promising approaches include the monitoring of sensor information, which is abundant on mobile devices, to detect abnormal communication [29], [30], abnormal physical parameters (e.g. temperature, cpu clock frequency) [31], or abnormal execution flow [32]. Information coming from all these monitors can be gathered in order to build advanced detection techniques [33]. As outlined in [10], several solutions rely on the observation of battery power. One of the proposed solutions, VirusMeter [34] monitors and audits power consumption on mobile devices with a behavioral power model that accurately characterizes power consumption of normal user behaviors. However, it remains an open research question, to what extent malware can be detected on smartphones in daily use, monitoring just the battery power, with continuously changing user behavior [10].

Depending on the point of view, an attack on a system can be seen as an anomaly or a failure. Detecting or predicting such events has been the body of research in different communities for a long time. This knowledge can be leveraged to adapt and further develop the existing methods. A comprehensive selection of works done in the field of anomaly detection can be found in [35], for failure prediction in [36] and, since we are classifying observed behavioral patterns, pattern recognition in [37], [38], [39].

D. Distribution of Attack Detection Algorithms

A model proposed in [40] consists of two components: a host agent and a network service. The main purpose of the host agent is to acquire files and send them to the network service, whereas the network service performs analyses using multiple detection engines in parallel to determine whether a file is malicious or not. Another proposed solution is ParanoidAndroid [41], that uses the anomaly detection principle. Based on phone execution traces, security checks are performed on the synchronised copy of the phone that runs on a server. Shortcomings of these approaches are their lack of support for behavioral detection and the delay of detection. Also, in both approaches data from the mobile side is being continuously sent into the network (both malicious and non malicious files) thus raising more privacy issues.

IV. Proposed Methodology

We propose a resource-optimized two-step malware detection methodology for mobile devices. In order to make the system more resistant to the increased number of malware threats, the methodology focuses on the identification of behavioral patterns known to relate to malware families, rather than to single malware samples.

The proposed system consists of a cloud service and a group of mobile devices, as shown in Figure 1. The methodology has the following procedure:

1) Mobile devices continuously run a local, efficient but less specific malware detection algorithm. They check for known families of malicious behavior. Upon detection, an alarm is raised on a device and sent to the cloud.
2) After receiving an alarm, specific, high precision detection algorithms are run to check for specific malware behavioral patterns within that family. The set of algorithms is chosen by the cloud service and is possibly supported by information from additional alarms raised by other devices. The specific algorithms are run on the cloud service using input data monitored on the mobile device. If the state of resources and the expected integrity of the device allow it, the cloud service may also delegate certain algorithms to be run on the devices themselves. In case one of the specific algorithms recognizes a malicious pattern, the device is considered to be infected by malware and appropriate countermeasures can be taken.

Algorithms to be used for malware detection will be selected from failure prediction methods, including machine learning algorithms.

Since a specific malware family can be detected with high precision in the cloud, based on previous knowledge about the family it is possible to estimate what damage it can cause and provide this information to the user. Additionally, if epidemic behavior is observed within the network a user can be notified about this, too. Under which conditions and how the user will be notified is out of the scope of this work, but has to be considered when deploying such a system.

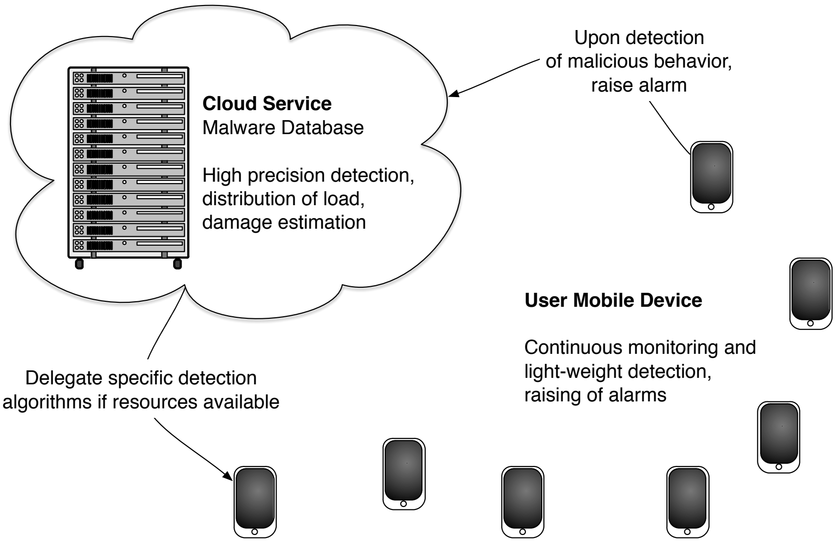


Fig. 1. Overall scenario for hybrid malware detection consisting of a cloud service and a set of mobile devices outside of the protected infrastructure.

One of the key issues is being able to minimize false positive and avoid false negative detections. Reaction latency is also a major issue as it is essential to stop malware before the system has been compromised or information has leaked. In the proposed two-step approach, a higher number of false positives is accepted in the first detection step with priority on detection recall. In the second step, high precision algorithms are run to eliminate false positives.

The methodology can be applied on any operating system. However, since the vast majority of all mobile malware in 2013 targeted Android devices [2], we propose its application on smartphones and tablets running the Android operating system as it has by far the largest install base worldwide and provides developers and researchers with sufficient freedom to carry out the foreseen tasks. In the following, we describe the individual steps of the proposed methodology in detail and propose solutions for each of them.

A. Behavioral Malware Database Preparation and Maintenance

As it provides the basis for all further steps, the creation of the malware database is described first. An overview of this part is presented in Figure 2. Modern mobile devices, such as smartphones, at their core seem to be no different than known multi-purpose computing architectures. But their specialization for specific use cases, coupled with the capabilities to handle the diversity of those use cases creates a new class of devices. As opposed to general-purpose PC architectures, most API calls to access and manipulate data imply strong security and privacy semantics. For example, the list of APIs on the Android operating system includes *Notification Listener*, *Contacts Provider* and *Localization*. Our methodology proposes to start by examining the device operating environments closely to properly describe their architecture from a security point of view and deduct possible malicious behavior.

The process is aided by investigating access patterns produced by known malware samples. The first step of the methodology is the development of a comprehensive database of malware samples and their behavior that goes beyond what can be found in [11], [42]. This is done by simulating existing mobile malware and monitoring how it affects the observed features on a mobile device. After simulation, malicious behavioral patterns are created in a similar way as in [16] where the logical ordering of an application's action over time is used instead of observing each action alone.

The aim of the methodology is to use a most comprehensive set of features (starting with the ones in [22], [43], [11], [20]) and select significant ones for behavioral patterns by evaluating precision and recall of the detection algorithms. The observed features may include traditional monitoring parameters provided by the hardware and operating system, such as the number of threads or the amount of memory usage. In addition, mobile devices provide features that denote access to specific sensitive data, like address books, photos or sensors like GPS or the camera. Finally, certain features provide information about network status or usage and more specifically, the social context of the phone and its user. The idea is to leverage the specific characteristics of a device to infer a set of malicious behavioral patterns.

The identified patterns of the observed features during simulation resemble the behavior of a specific malware sample. Sets of malware samples that induce identical behavioral patterns are considered *malware families*. We call these family specific behavioral patterns *specific behavioral patterns*. Simplified behavioral patterns, including less features, are also extracted. We call them *generic behavioral patterns*. Behavioral patterns are separated into family-specific and generic ones in order to provide 1) good detection precision of malware families at the cloud side, 2) an efficient algorithm that is able to detect suspicious behavior at the device side.

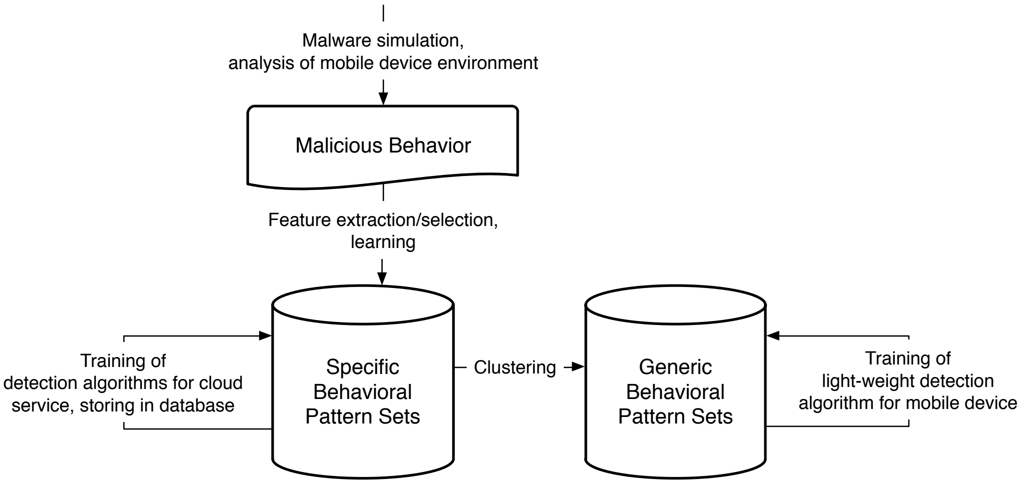


Fig. 2. Generation of specific behavioral patterns database (left), generic behavioral pattern sets (right) and their respective detection algorithms.

After classification, the most indicative features for the detection algorithms of each pattern set are selected, considering the methods such as [18]. This is done separately for the algorithm that is run on the device and the set of algorithms run on the cloud service. Regarding the latter, for each set of specific behavioral patterns, an optimized detection algorithm is defined. This algorithm, including its configuration and an estimation about its resource requirements is collected in the database.

The existing malware database is updated upon detection of new malware samples. New samples are simulated and their respective behavioral patterns checked in order to decide if the system is still working with the most representative features. This is done in order to keep an optimal balance in between of type of features, their number and precision.

B. Run-Time Malware Detection

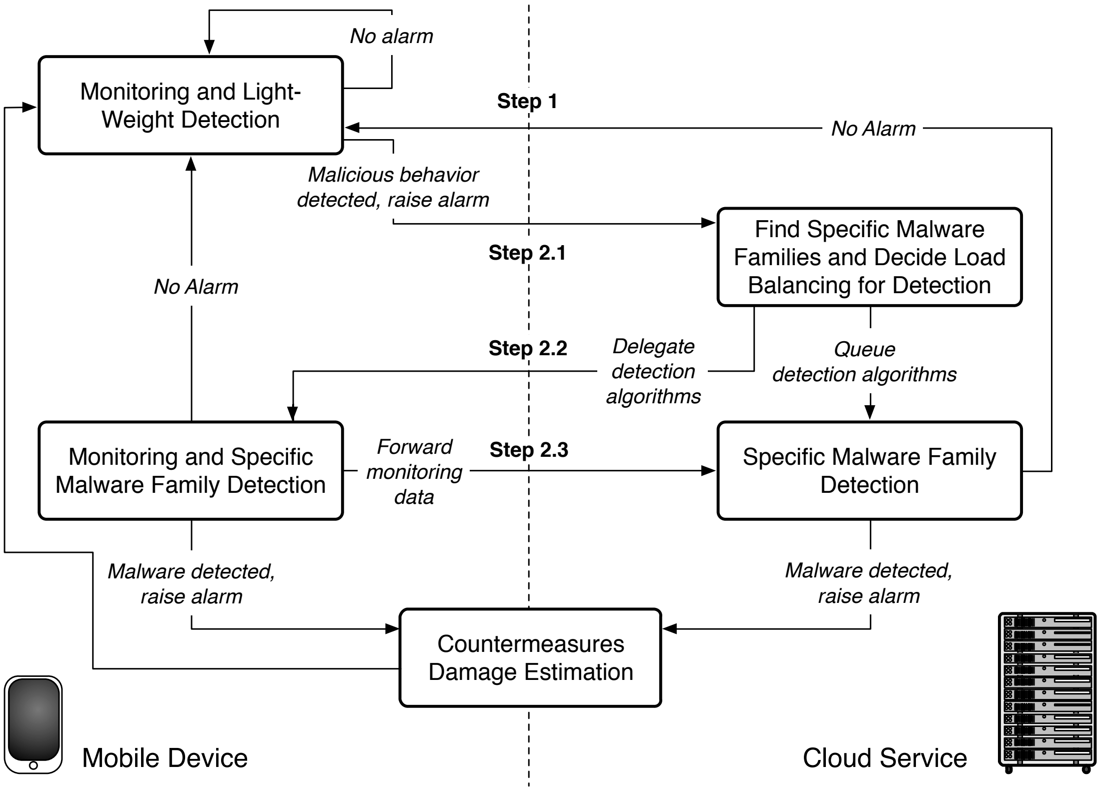
During operation, all devices provide an indicator about their security state: green during regular operation, yellow when a generic alarm has been raised until all specific pattern detection algorithms have been run without result and red in case a specific algorithm finds a match. This state is known to the cloud service for all mobile devices and can be used for further analysis, e.g. epidemic analysis of malware propagation or statistical analysis like the *mean time to attack*.

In the following, a step-by-step description of the two-step approach for malware detection is presented, as illustrated in Figure 3. The challenges and proposed solutions to be developed are also discussed.

1) Step 1: On-Device Light-Weight Malware Detection: Basis of the approach is a continuous monitoring and malware detection on each mobile device, in an attempt to detect generic behavioral patterns that match known malicious behavior. In order to reduce the number of monitored features and keep the algorithm run on the device efficiently, the algorithm identifies only generic patterns. The number of features to be considered is a result of a multi-objective optimization between precision and recall on one side and power consumption and complexity on the other. In order to investigate any detected malware family in a further step, a high recall has priority.

Upon detection of a malicious generic behavioral pattern, an alarm is raised. Due to the optimization for resource requirements and recall, precision is expected to be lower. This is expected to manifest itself in a fair number of false positives. For further investigation of the alarm, the behavioral pattern that led to the alarm is sent to a cloud service together with the state of currently available resources on the device (i.e. battery, cpu, memory, network). Different algorithms, such as Naive Bayes Classifier [44], [45], the Universal Basis Function [46] and Support Vector Machines [16] are investigated as potential candidates for on-device pattern recognition.

2) Step 2.1: Matching Suspicious Behavior to Known Malware: Upon receiving an alarm from the mobile device at the end of Step 1 means that behavior common to a malware family was detected. The cloud service then checks the alarm data to verify whether it can classify the detected pattern into one of the specific behavioral patterns. The list of specific behavioral patterns is found in the malware database. Found specific patterns are ranked by their probability, the ranking process is supported by alarm data arriving in the same time interval from different mobile devices. For each of the specific behavioral pattern sets (hence, malware families patterns), there is an optimized recognition algorithm. The main goal of these algorithms is to detect and identify specific malware. Additionally, since they are run on the cloud service, resource limitations are of minor concern. More complex algorithms. which are optimized for F-measure, the combination of precision and recall, are run. These algorithms are, for example, Hidden Semi-Markov Models [47] and neural networks [27], [38]. For each pattern, the best one based on its achieved confidence level is chosen. After optimization, each specific behavioral pattern has a customized recognition algorithm and a number of representative features. Furthermore, the



No alarm
Monitoring and Light-Weight Detection
Step 1
No Alarm
Malicious behavior detected, raise alarm
Find Specific Malware Families and Decide Load Balancing for Detection
Step 2.1
No Alarm
Step 2.2
Delegate detection algorithms
Queue detection algorithms
Monitoring and Specific Malware Family Detection
Forward monitoring data
Step 2.3
Specific Malware Family Detection
Malware detected, raise alarm
Malware detected, raise alarm
Countermeasures Damage Estimation
Mobile Device
Cloud Service

Fig. 3. Overview of the hybrid approach for runtime malware detection.

resource requirements for each of the optimized algorithms are estimated. The database thus holds a specific, highly precise recognition algorithm configuration for every specific behavioral pattern set with an estimation of the duration of such an attack and the resource requirements to run it.

3) Step 2.2: Resource Allocation for Detection Algorithms: Having a list of potential malware that may have caused the alarm in Step 1, the cloud infrastructure has to make runtime decision on where the detection algorithms for each malware will run, on the device itself or in the cloud. This decision is the result of a multi-objective optimization among the requirements for the algorithm, the available resources on the device (i.e. battery, cpu, memory, network) and an estimation of the state of compromise of the mobile device. To support the latter objective, an evaluation of the expected damages over time by the different malware is needed.

4) Step 2.3: Specific Malware Detection: After the decision in Step 2.2 about where to run each detection algorithm, they are executed with the goal to recognize a specific behavioral pattern set. Detection is carried out for an amount of time corresponding to the expected duration of the assumed attacks. In case an algorithm is run on the mobile device, the phone informs the cloud service only about the outcome of the algorithm. If an algorithm is run in the cloud, the mobile device forwards the monitored features to the cloud service. Communication channels between the mobile devices and the cloud are protected. If no clear classification can be done by any of the recognition algorithms, the alarm is considered as a false positive and the mobile device falls back to regular operation in Step 1. However, when the confidence of the malware detection is high enough, either on the device or on the cloud, a specific alarm is raised that is expected to have a very high probability to be correct. Appropriate countermeasures with respect to the damage estimation can then be employed.

V. Comparison of the Methodology with State of the Art Solutions

In [16], one possible representation of malware behaviors together with the generation of a database of malicious behavior signatures of different mobile malware families is described. Unfortunately, to the best of our knowledge this method was used only for Symbian OS. The methodology we propose builds on this findings. It starts from applying a similar approach in order to create a database of different Android OS malware family behavior.

While existing anti-virus techniques are expected to be partly effective in addressing smartphone viruses, there are also several limitations on mobile devices, mainly related to resource constraints, battery life in particular. Additionally, *polymorphic* viruses, which continually rearrange their code to evade detection, are also being developed for mobile systems. The one way to detect polymorphic viruses is to look for virus-like behavior. Smartphone software does not have the sophistication to detect these viruses, due to resource limitations [48]. Our methodology addresses these shortcomings by applying behavioral detection algorithms supported by a cloud infrastructure.

The proposed methodology is more similar to SmartSiren [49], where each smartphone runs a lightweight agent, while a centralized proxy is used to assist the virus detection and alert processes. In SmartSiren, each smartphone reports a summary of communication activities to the proxy, periodically and upon detection of abnormal activities. The proxy then performs a joint analysis on the received reports and detects single device or system wide viral behaviors. However, SmartSiren only uses Bluetooth and SMS communication to detect unknown behavior and is focused just on virus detection. More comprehensive methodology than SmartSiren, such as the one we are proposing, is needed. It should include features selected by statistical methods and related to overall system behavior and to consider different detection algorithms in both a device level and cloud side in order to find the efficient ones with respect to available computational resources and precision.

VI. Impact and application to other areas

The proposed methodology can be applied to environments where connectivity during regular operation is necessary but the integrity of devices is of crucial importance, such as, corporate smartphone deployments. In such a scenario, there is an existing protected, corporate network infrastructure to which mobile devices can connect via virtual private networks. Outside the infrastructure, a mobile device is considered to be constantly in a hostile environment. There, a mobile device can be attacked by malicious third parties to steal corporate secrets or to hinder communication, to gain an advantage over a competitor or to disrupt its operation. A compromised device will further endanger every other device within the corporate infrastructure.

By using behavioral detection, a system is more resistant to new malware. While attackers, by simple obfuscation of existing malware samples, can bypass binary signatures based detection, to bypass a system that is based on behavioral detection of malware families, the creation of a new malware family is needed, which requires much more effort.

One of the most important properties of the methodology is that it addresses the limitations of mobile devices by distributing the behavioral detection algorithms among the mobile device and a cloud service. It takes into account the computational resources of a mobile device and, based on that, proposes appropriate lightweight mechanisms to be run on it. That makes it suitable for a variety of mobile devices, starting from older mobile phones with limited computational resources towards new smartphones and tablets with more computational power.

Additionally, we foresee that the methodology can be used also in other embedded systems to protect them against threats. According to Cisco's research in [2], malware threats toward electronics manufacturing, the agriculture and mining industries are increasing. Although in different scenarios behavioral patterns can be different, the two steps approach is still applicable.

VII. Conclusion

A methodology for early detection of malware in mobile devices was presented. The main contributions of the proposed methodology are in behavioral malware detection of malware families rather than single malware samples, continuous update of a system if new malware families appear and resource-optimized, distributed malware early detection taking place, depending on complexity, on the mobile device or the cloud infrastructure. Additionally, the system should be able to give the precise description of detected malware family, estimate potential damage to the phone and suggest appropriate countermeasures to be taken.

References

[1] (2013, November). [Online]. Available: http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-november-2013.pdf

[2] (2014). [Online]. Available: https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf

[3] "2014 threats predictions," McAfee Labs, Tech. Rep., 2014.

[4] (2012, February). [Online]. Available: http://www.securelist.com/en/analysis/204792222/Mobile_Malware_Evolution_Part_5

[5] J. Viega and H. Thompson, "The state of embedded-device security (spoiler alert: It's bad)," *IEEE Security and Privacy*, vol. 10, no. 5, pp. 68–70, 2012.

[6] K. Dunham, *Mobile Malware Attacks and Defense*. Elsevier Science, Syngress, 2008.

[7] G. Delac, M. Silic, and J. Krolo, "Emerging security threats for mobile platforms," in *MIPRO, 34th International Convention*. IEEE, 2011, pp. 1468–1473.

[8] C. Guo, H. J. Wang, and W. Zhu, "Smart-phone attacks and defenses," 2007.

[9] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[10] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C. Wolf, "Mobile security catching up? revealing the nuts and bolts of the security of mobile devices," in *Symposium on Security and Privacy*, ser. SP '11. IEEE Computer Society, 2011, pp. 96–111.

[11] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *1st ACM workshop on Security and privacy in smartphones and mobile devices (SPSM)*. ACM, 2011, pp. 3–14.

[12] S. Khan, M. Nauman, A. Othman, and S. Musa, "How secure is your smartphone: An analysis of smartphone security mechanisms," in *International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012, pp. 76–81.

[13] N. J. Percoco and S. Schulte, "Adventures in bouncerland, failures of automated malware detection within mobile application markets," Trustwave Holdings, Inc., Tech. Rep., 2012, black Hat Convention.

[14] C. Yavvari, A. Tokhtabayev, H. Rangwala, and A. Stavrou, "Malware characterization using behavioral components," in *Computer Network Security*, ser. Lecture Notes in Computer Science, I. Kotenko and V. Skormin, Eds. Springer, 2012, vol. 7531, pp. 226–239.

[15] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 6:1–6:42, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/2089125.2089126

[16] A. Bose, X. Hu, K. G. Shin, and T. Park, "Behavioral detection of malware on mobile handsets," in *6th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2008, pp. 225–238.

[17] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[18] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[19] G. W. Chow and A. Jones, "A framework for anomaly detection in okl4-linux based smartphones," in *6th Australian Information Security Management Conference*. Security Research Centre, School of Computer and Security Science, Edith Cowan University, 2006.

[20] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *16th ACM conference on Computer and communications security (CCS)*. ACM, 2009, pp. 235–245.

[21] A. P. Felt, K. Greenwood, and D. Wagner, "The effectiveness of application permissions," in *2nd USENIX conference on Web application development (WebApps)*. USENIX Association, 2011, pp. 7–7.

[22] A.-D. Schmidt, F. Peters, F. Lamour, and S. Albayrak, "Monitoring smartphones for anomaly detection," in *1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, ser. MOBILWARE '08. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 40:1–40:6.

[23] H. T. T. Truong, E. Lagerspetz, P. Nurmi, A. J. Oliner, S. Tarkoma, N. Asokan, and S. Bhattacharya, "The company you keep: Mobile malware infection rates and inexpensive risk indicators," *CoRR*, vol. abs/1312.3245, 2013.

[24] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models," in *Symposium on Security and Privacy*. IEEE, 1999, pp. 133–145.

[25] X. Kou and Q. Wen, "Intrusion detection model based on android," in *Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on*, 2011, pp. 624–628.

[26] M. Miettinen, P. Halonen, and K. Hatonen, "Host-based intrusion detection for advanced mobile devices," in *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on*, vol. 2, 2006, pp. 72–76.

[27] A. Boukerche and M. S. M. Annoni Notare, "Behavior-based intrusion detection in mobile phone systems," *Journal of Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1476 – 1490, 2002.

[28] A. Houmansadr, S. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *Dependable Systems and Networks, Workshops (DSN-W)*. IEEE/IFIP, 2011, pp. 31–32.

[29] J. Coburn, S. Ravi, A. Raghunathan, and S. T. Chakradhar, "Seca: security-enhanced communication architecture," in *International conference on Compilers, architectures and synthesis for embedded systems (CASES)*, T. M. Conte, P. Faraboschi, W. H. Mangione-Smith, and W. A. Najjar, Eds. ACM, 2005, pp. 78–89.

[30] P. Cotret, G. Gogniat, J.-P. Diguet, and J. Crenne, "Lightweight reconfiguration security services for axi-based mpsocs." in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, D. Koch, S. Singh, and J. Tørresen, Eds. IEEE, 2012, pp. 655–658.

[31] T. Wolf, S. Mao, D. Kumar, B. Datta, W. P. Burleson, and G. Gogniat, "Collaborative monitors for embedded system security," in *1st Workshop on Embedded Systems Security (EMSOFT)*. ACM, 2006.

[32] S. Mao and T. Wolf, "Hardware support for secure processing in embedded systems," in *IEEE Transactions on Computers*. IEEE, 2010, vol. 59, no. 6, pp. 483–488.

[33] B. Robisson, M. Agoyan, S. Bouquet, M.-H. Nguyen, S. Le Henaff, P. Soquet, G. Phan, F. Wajsbürt, P. Bazargan-Sabet, and N. Drach-Temam, "Management of the security in smart secure devices," in *International Conference on Smart System Integration (SSI)*, 2011.

[34] L. Liu, G. Yan, X. Zhang, and S. Chen, "Virusmeter: Preventing your cellphone from spies," in *12th International Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer, 2009, pp. 244–264.

[35] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[36] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Computing Surveys*, vol. 42, no. 3, pp. 10:1–10:42, 2010.

[37] N. J. Nilsson, "Survey of pattern recognition," *Annals of the New York Academy of Sciences*, vol. 161, no. 2, pp. 380–401, 1969.

[38] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Schölkopf, Eds. Springer, 2006.

[40] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," in *1st Workshop on Virtualization in Mobile Computing*, ser. MobiVirt '08. ACM, 2008, pp. 31–35.

[41] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid android: versatile protection for smartphones," in *26th Annual Computer Security Applications Conference (ACSAC)*. ACM, 2010, pp. 347–356.

[42] A. Nappa, M. Rafique, and J. Caballero, "Driving in the cloud: An analysis of drive-by download operations and abuse reporting," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. Lecture Notes in Computer Science, K. Rieck, P. Stewin, and J.-P. Seifert, Eds. Springer, 2013, vol. 7967, pp. 1–20.

[43] Y. L. Ho and S.-H. Heng, "Mobile and ubiquitous malware," in *7th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM '09. ACM, 2009, pp. 559–563.

[44] A. Csenki, "Bayes predictive analysis of a fundamental software reliability model," *IEEE Transactions on Reliability*, vol. 39, no. 2, pp. 177–183, 1990.

[45] M. Pizza, L. Strigini, A. Bondavalli, and F. di Giandomenico, "Optimal discrimination between transient and permanent faults," in *3rd International High-Assurance Systems Engineering Symposium (HASE)*. IEEE, 1998, pp. 214–223.

[46] G. A. Hoffmann, "Failure prediction in complex computer systems: A probabilistic approach," Ph.D. dissertation, Humboldt-Universität zu Berlin, 2005.

[47] G. A. Hoffmann and M. Malek, "Call availability prediction in a telecommunication system: A data driven empirical approach," in *25th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2006, pp. 83–95.

[48] C. Biever, "Phone viruses: How bad is it?" Article, NewScientist, 2005. [Online]. Available: http://tinyurl.com/qy2jawu

[49] J. Cheng, S. H. Wong, H. Yang, and S. Lu, "Smartsiren: virus detection and alert for smartphones," in *5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. ACM, 2007, pp. 258–271.