Lightweight Forensics Application: Lightweight Approach to Securing Mobile Devices

Helen Angela Brumfitt
Liverpool John Moores University
Department of Computer Science
Liverpool, UK
H.A.Brumfitt@2008.ljmu.ac.uk

Bob Askwith
Liverpool John Moores University
Department of Computer Science
Liverpool, UK
R.J.Askwith@ljmu.ac.uk

Bo Zhou
Liverpool John Moores University
Department of Computer Science
Liverpool, UK
B.Zhou@ljmu.ac.uk

***Abstract*— Physical objects with the addition of sensors, actuators and a connection to the internet form devices which can collect, process and communicate data to each other. Devices may not have been designed with connectivity in mind and adding it as an afterthought is problematic. This provides a significant technical challenge concerning securing the devices, as they are all of a sudden open to a wide range of attacks whilst providing more opportunities for malicious users and increases the chances of device compromise. The key aim of this research is to address limitations in current security solutions on mobile devices by defining a novel approach which will sustain future advances in mobile technology. Using combined security techniques our proposed solution will work with existing security technology to create a more effective and successful security implementation that will be suitable for a wide range of mobile devices.**

Keywords- Lightweight security; mobile device; smartphone; digital forensics; malware detection; in-network; Collaborative; Internet of Things

I. INTRODUCTION

Integrating with our daily lives mobile devices are becoming more valuable to us than the personal computers we appear to be leaving behind. The opportunities ubiquitous computing provides us with are endless, from healthcare to military applications as well as enterprise and personal home or office networks. Due to this embrace of technology, ubiquitous computing has influenced the advanced technology now available, inevitably making us expect more to be possible in the future such as the smart cities [1], homes and the Internet of Things [2] we have been promised.

Mobile devices have evolved to be extremely capable devices augmenting and enhancing control of our daily lives. Ubiquitous in nature and containing CPU, memory, input and output abilities and storage makes them equal with computers from several years ago. These devices allow users to complete a number of tasks including browsing the internet, sending and receiving emails, installing apps, a portal to their documents on the cloud, storage, streaming movies, downloading music and a camera are amongst some of the well-known capabilities. This new paradigm will gradually fuse the digital and physical worlds we currently dwell in and open up many new opportunities for our professional, personal and social environments, opening paradigms such as wearable computing and the Internet of Things [2].

Environments are formed by these heterogeneous devices which can potentially have low processing power and low memory capabilities but still interact with each other through wireless networks. When working with a heterogeneous infrastructure such as this security can be overlooked. As smartphones run operating systems that are similar to a traditional desktop PC, they can also suffer from the same kind of weaknesses and vulnerabilities as them as well as new ones [3]. Android is statistically the most targeted platform by cyber criminals. More than 99% of new mobile malware in 2014 was aimed at Android devices. There were 727,790 installation packages, 65,118 new mobile malware programs, 2033 mobile banking Trojans in the second quarter, this was lower than the first, although has been put down to the holiday season [4]. An environment of connected mobile devices is the key to innovation in our future. However this is now forcing us to consider the security and privacy concerns it is presenting as some of these devices can carry a great deal of sensitive data. Mobile devices can see, hear and sense their environments which makes them a favourite target to malicious users because of their data centric approach and complex systems making security difficult to implement. This puts our trust, privacy and security on the devices at risk [5].

In this work we present our contribution in the form of a mechanism which forms part of our novel security framework, consisting of three components. Each component is designed to enhance existing security solutions using both existing techniques and new mechanisms. An outline of the framework was first presented in [6], although in this paper we focus on one component. The rest of the paper is organised as follows. Section II presents an overview of mobile device security. We expand on this in section III and look at existing security and research. We introduce our framework in section IV followed by the implementation of a specific component in section V. We will then discuss this in section VI, followed by a conclusion and further works in section VII.

II. MOBILE DEVICE SECURITY

Our desire for innovation, slowly fusing the digital and physical worlds together is unfortunately fuelling the payload an attacker can achieve through a compromised device. As highlighted in [7] the number of attack vectors are multiplied on mobile devices, as they have so much more to offer for the attacker. The cost of initiating an attack will be less than the potential revenue they will receive if it is successful. Further more, users now also store passwords and account information on mobile devices as well as the increased use of Near Field Communication (NFC) which saves a users financial information.

From this we can deduce that the introduction of mobile devices in general, not just smartphones alone have impacted on our security in ways which PCs couldn't do. As our devices become more sophisticated, the attacks which can be used to compromise them also grow more sophisticated. In the next section we will look at the different ways in which an attacker can compromise a mobile device, as well as review some of the motivations behind attacks.

A. *Challenges of Securing Modern Networks*

Smartphones are so ubiquitous and heterogeneous they are becoming complex systems that are difficult for network managers to manage safely in the network and also difficult for users to manage their own privacy. Some of the challenges are discussed below [8].

- Mobile devices have an increased number of vectors in which they can be infected, including by Bluetooth, MMS, HTTP and SMS or generally through attacks in the application layer, communication protocols and operating system.
- Mobile devices are usually always on and with the user. Alongside numerous sensors present on the device, this could potentially allow the device to continuously sense the context of a users environment.
- Smaller devices means potentially less resources, CPU and memory. This results in heavy security software being unable to run or not as efficient on the devices.
- Constant movement between numerous unknown networks. As most computing devices are now mobile, users can connect them to numerous networks which may or may not be trusted, and send sensitive data through the network.
- Some users do not seem to know much about these advanced devices and how to use them safely and securely. They may apply the same password for everything or download apps which are not verified without checking the permissions the app requests.
- Low physical protection surrounding a mobile device. Devices like smartphones and tablet computers can easily be stolen or left at different locations.
- Contradicting goals of current security such as the requirement for security but without it directly affecting the usability of the device can limit the security applied.
- The Bring Your Own Device (BYOD) policies have a compromise in the way they can't control what a user uses their device for and what they put on their device as well as the security they run on it.

III. EXISTING SECURITY

Security on mobile devices can be looked at as either on the device or implemented at market level. In the application market, apps may be subjected to a review in which it will be analysed and tested against a set list of criteria. Application signing is also used in which each app published must be signed by the author in order to establish authorship. However even with these two security features in place attacks can still compromise the device. On device security includes the use of permissions which is designed to restrict what actions an app can perform on the device, although in order to get round this attacks have been known to use multiple apps in order to achieve a goal. Users also grant all the permissions requested by an app therefore allowing it to do what it wants.

Google has a method of detecting malicious apps before they are integrated into the Google Play Store [9]. Their solution, Bouncer, is able to determine if apps send SMS messages out to malicious users, although this technique isn't useful to those users who take advantage of the third party app stores. Samsung also have their own security system known as Knox in their new line of smartphones. Their strategy includes secure boot, ARM TrustZone-based Integrity Measurement Architecture, and a kernel with built in security measurements. This creates a good starting point to build appropriate security that is integrated into the mobile devices [4].

In [10] the authors address mobile security challenges and develop Mobile Guardian, a framework for security policy enforcement on mobile devices. In particular they investigate sensitive data isolation, security policy formulation, security policy testing and security policy execution. Their framework is designed to be secure, flexible and scaleable whilst targeting enterprise networks over personal networks due to the BYOD policies. It can be adopted on many platforms in order to implement access control, data confidentiality, security and integrity.

An end-to-end security framework is introduced in [11]. It focuses on the specialised web transfer Constrained Application Protocol (CoAP) as it provides strong security whilst running on the smallest of nodes and networks with constrained resources. In their work they bind the CoAP with Datagram Transport Layer Security (DTLS) to provide end-to-end security without incurring much overhead. This is similar to the research in [12] which again utilises the CoAP protocol but this time alongside an optimised implementation of the Elliptic Curve Digital Signature Algorithm (ECDSA) inside a smart object. In this work the authors focus on scenarios such as in e-health and smart buildings where vast amounts of sensitive data are managed and if leaked could harm the privacy of the users. The authors design a distributed approach to control access to the sensitive information on such networks which involves the smart devices themselves making fine-grained authorisation decisions.

Current security is not well adapted and needs to be enhanced by new thinking and not just small solutions filling gaps but by a new integrating 'framework' way of approaching it. This is recognised in other works such as [13] in which the author identifies the need for a new secure system architecture or re-evaluating and enhancing the existing architecture so it will be suitable for deployment on new networks such as the IoT. Security would benefit by being enhanced in a way that will allow the same framework to be applicable in numerous applications in different environments, both utilising existing techniques as well as new ones. This is more important now as technology is constantly evolving at an immense rate, we don't know and have no way of knowing what could be

released within the next ten years. Keeping up with the fast pace of technology and its prevailing threats is a challenge in itself as is trying to keep ahead of malicious threats, attacks and vulnerabilities. In order to overcome these challenges we have identified a solution that neither replaces the full security or sections of it. It adapts current security in a way that it will enhance it for future networks. In this work we focus on one particular component of our framework, although in the next section we will introduce the full framework to see how this component fits in.

IV. Framework Solution

In order to address the challenges identified above, we have designed a novel framework which is aimed at enhancing general security as well as providing stable building blocks to develop future security. Our framework consists of three components; 1) a Lightweight Forensics Application (LFA) that runs on the device 2) a Central Security Manager (CSM) that runs in the network and 3) a collaborative component that will run between devices in a network. The three components are each designed to complete their own tasks, however the three of them are integrated and work with each other in order to create an effective framework for security. The framework itself is designed to be applicable on any type of network including IoT, cloud, home, enterprise, public or specialised such as medical networks. The concept of the framework will not change despite the environment it is in. The integration of the three components can be seen in figure 1. Both the LFA and Collaborative (Col) components collect data for the CSM. In this section we introduce each of the components and briefly what their individual tasks are.

A. *Lightweight Forensics Application (LFA)*

The LFA is designed to be lightweight to ensure it can be dynamic to adapt to any device it will be integrated on. This includes devices with very low processing power such as small sensors. It has a priority to collect data that the CSM can use in order to enforce security. It does this by adhering to a predefined set of classifications. This data is then sent to the CSM component for further analysis. The LFA doesn't need to know why it is looking for specific data. It has been inspired by forensics as it has a set number of actions to look for but does not know specifically what it is looking for. We are assuming the LFA will run in a protected kernel within the device itself, although saving minimal data and none about the user makes it a low target for malicious users. The LFA may respond to one CSM in work, and another at home, allowing the companies to specify policies and implement them.

B. Central Security Manager (CSM)

The CSM contains all the heavy processing and decision making. The CSM collects data from the LFA and Col components. Resources and processing power will be required in order to utilise data being sent to it from the LFA and Col components. Placing the CSM in the network therefore ensures it has access to what it needs, as well as other assets such as firewalls and Intrusion Detection Systems (IDS) or Anti-Virus. The CSM is designed to be highly automated, making decisions for itself rather than waiting for an admin to assess the problem. Reports will be created by

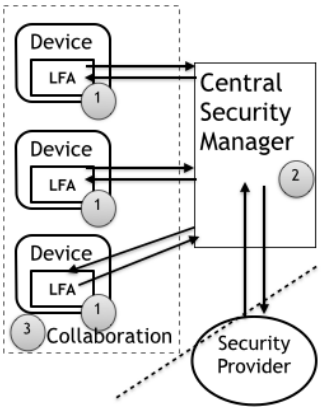

Fig. 1. Framework Component Integration

the CSM so security administrators can be kept in the loop of what is happening, and some events may require their attention.

C. Collaborative Component

The Col component collects data in the distributed network which can indicate possible vulnerabilities, threats and attacks on a larger scale. This component also allows for different policies to be applied in different networks. For example a user may have a personal Col network at home with family devices connected to one CSM. As they arrive in work their device automatically switches to the business Col network to ensure policies are followed. This will ensure devices are useable at home as normal although in work restrictions may take hold in order to follow the corporate policies for BYOD. The Col component also allows us to further ensure the LFA remains lightweight by including shared processing and tasks as well as ensuring the LFA on each device is not compromised and secure.

V. Lightweight Forensics Application Component

We have briefly introduced each of the components and how they integrate with each other. We will now look into their design further to show how they can enhance current security.

A. Lightweight Forensics Application Design

It could be that in the future that the LFA will facilitate the collection of the data for other purposes such as medical applications, however we are only concerned with the data collection in this work for the CSM. In order to implement this we have decided the LFA has the following main mechanisms:

App Runtime - We are going to assume that the LFA will run as a module in protected kernel space within the mobile device, therefore not accessible via application or operating system vulnerabilities. The LFA quietly monitors actions on the device and will trigger an event when a condition of one or more of the classifications are met. It doesn't store any user data, and will only store triggered events for a short period of time. Purely a data collection component this reduces the risk of it becoming a target for compromise.

Classifications - We don't want the LFA to be searching for everything all the time as it will use too many resources continuously on constrained devices. Generally an Intrusion Detection System (IDS) uses either anomaly detection or misuse detection to identify malicious actions on a system. In our work, we will be using neither of these as they are too resource intensive for most smaller mobile devices to maintain. We use our own novel technique which will prove lightweight and more efficient to our needs. Our mechanism is formed using classifications, which minimises the list of events the LFA is monitoring for at any one time. This method also ensures low false positive events and aids the LFA in prioritising certain trigger events.

The classifications are created and provided by the security administrator initially. Classifications can be used to detect general events or specific threats or events depending on what the CSM or the admin deems necessary. Different classifications may be used on separate devices depending on the risks on the device. For a smartphone used for business the security administrator may choose to have classifications which detect data leakage, unintentional disclosure of data and surveillance attacks. For a personal smartphone the user may choose to have classifications which can detect spyware, financial malware or diallerware attacks. In a smart home environment the classifications could include personal security and surveillance. Once the LFA detects actions relating to these classifications it will send reports to the CSM. The CSM may make the decision then to alter the classifications to allow it to gain more information if it suspects anything else that could be happening on the device. The CSM may also obtain data from external inputs such as IDS or firewalls and utilise the data to create new classifications for the connected LFA's to follow.

Threats which have not yet been seen before could be recognised as a result of this classification method although the LFA will not recognise it as a new threat, this will be down to the CSM. Using this mechanism, we can ensure the LFA remains as lightweight as possible, whilst also not missing suspicious events.

Trigger Events - When a suspicious event is detected on the device, a report will firstly be sent to the LFAs own event log system. If the event has a high priority then it will be sent by the LFA to the CSM immediately to ensure it is analysed quickly. If the event is not high priority then the LFA will leave it saved within its own memory until the CSM requests it. To maintain security and ensure there is no compromise the CSM will randomly request an event report from the LFA. This will be approximately every 24 hours. Each event will be uniquely numbered and encrypted with a one time cypher so the CSM can check against previous reports that it hasn't missed any requested reports and it can also determine if a report has been compromised and tampered with. Sorting the reports in terms of priority also ensures the CSM is using its resources more effectively. Once the CSM receives a valid event report it will send a receipt to the LFA which will then delete all the event reports it has previously sent. This will ensure the LFA does not start using too much storage space within the device and reduces the amount of data that could be extracted if the device was compromised.

B. Experimentation and Results

In this remainder of this work we will discuss the experimental implementation we have undertaken and the results we have achieved so far. Due to its timeliness we have decided to use the Internet of Things as a platform on which to trial the LFA. As the smartphone is a very common device within the IoT we will utilise it for our experiments. Although the theory of the LFA and ultimately the framework itself will allow it to be implemented in various environments.

In order to experiment with the classification mechanism we produced, we have used the Android Eclipse device emulator. One of the test classifications we developed was targeting SPAM messages to a smartphone. If the LFA follows the classification and detects the words "download", "install" and "http://" it automatically notifies the user of the detected possible SPAM message. We experimented with the classification method for a number of different types of example SPAM and social engineering messages and these were picked up by our classification method as expected whilst ignoring all normal social messages sent. This ensures our classification method is suitable for this framework.

Some classifications such as the SMS monitor may run continuously on the device, whereas other classifications may only run intermittently. In particular we have used the alarm feature in Android eclipse which runs a background service at regular intervals relying on an alarm to initiate it. This ensures the resources used are as low as possible. In an experiment we have created it checks the battery level every two hours, and if it has dramatically increased compared to the last time it has checked it will check the time of day it is. If it is night time and the phone is in idle mode, a trigger event will be created.

Figure 4 shows the memory usage of the SMS monitor. The Android Eclipse DDMS information console provided the data. The majority of the memory shown is free memory. The section with the arrow pointing to it and the relevant label is the SMS monitor and is significantly low. In order to experiment with this further we added on more classifications for the LFA to be looking for. In particular it monitored for incoming and outgoing calls, the

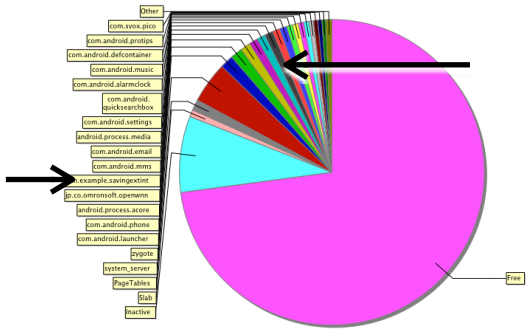

Fig. 4. Memory usage of SMS monitor

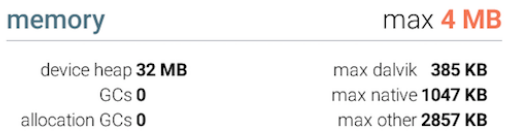

Fig. 5. Memory Usage of the modified SMS monitor.

Fig. 6. CPU usage of the modified SMS monitor.

state of the microphone and camera and also implemented the battery classification discussed earlier.

Whilst the modified SMS monitor application was running on a physical device we again monitored memory it was using. The results can be seen in Figure 5. We are encouraged by the reduced memory it is using at it is only searching for minimal events due to the classification mechanism, even though it is monitoring for a number of items. We are also confident we can reduce this more, as it is still currently running on the screen of the device all the time. The CPU usage for this app can be seen in Figure 6, throughout running the app and purposely triggering a number of the events the CPU % stays below 10%. As discussed earlier some of the classifications may continue to run all the time, whilst others may only run at specified points during the day or night. This will ultimately help to keep the resources used to a minimum.

In order to communicate this event report with the CSM we needed an output method. We initially instructed the LFA to save the number and text which triggered the classification to a .txt or .csv file internally on the device. This represents the report and log created by the LFA that would be sent to the CSM. We have implemented this on the virtual device emulated by Eclipse. We have also implemented it on physical Android devices although we had to integrate a way to save it to an external SDCard. This is due to being unable to access files on the device directly, which proves a problem for the testing point of view but in a final LFA implementation would not remain an issue.

The trigger event report is an important feature as it must

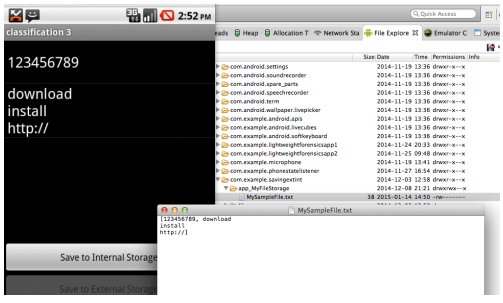

Fig. 7. The output file received from SMS monitor.

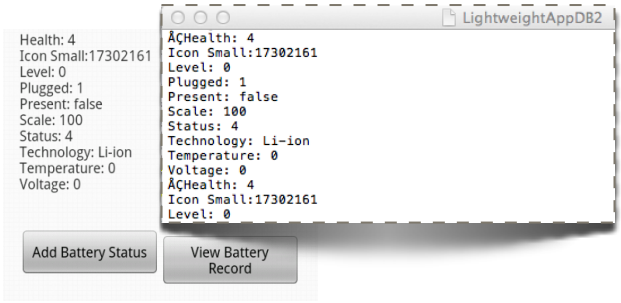

Fig. 8. Saving Data in an SQLite

contain all the information the CSM requires in order to make it useful, whilst also ensuring it doesn't over complicate matters for the LFA causing it to use more resources and processing power trying to send it.

We have experimented using an output as .txt and .csv files, as these can then be imported into a database. The CSM will be able to use this feature to create a log of all the events, separate them into relevant classifications and look at them all as an overview to discover patterns in trigger event behaviour across the device. The CSM will then create a full report which can be reviewed by relevant specialists who can identify the malicious actions taking place on the device. The CSM may also decide to request a sample, for example coding of an app, the permissions it uses, what data it gathers so it can integrate this into the report and send it to specialised scanning software to identify known malicious software.

Figure 7 shows the output file we have received from classification 3. At the moment it only shows the output to include the phone number it was received from and also the event that triggered the LFA to react to it. Realistically this would include more data regarding the classification without using any personal user data. In further experiments we decided on using SQLite files to save the events detected on the device. This can be seen in figure 8 using the battery classification. This decision was made as it allows us to then add further data in as required in the same file and makes it easier for the CSM to read. The LFA can easily delete the records after they have been sent to the CSM to ensure they don't take up too much memory.

VI. DISCUSSION

In this paper we have explored alternatives to current security by complementing and extending existing security in the form of a novel framework. A combination of which we believe will be a significant step to enhance the security of our devices from malicious actions in future networking. This work has focused on one particular component within our framework, the LFA. This work has successfully set out to prove that the theory of the LFA is feasible within a series of experiments and components and will complement existing security. Our results in this work are very encouraging and the LFA can be adapted onto numerous devices with small or large computing abilities.

Current and future networks are moving towards heavy processing and data storage on the network, and smaller devices on the edge of the network. These devices can include a range of types such as smartphones, desktop computers, laptops, smart watches, smart fridges, smart TVs and even smart houses and may be restricted in terms of the resources they have. Our framework is a perfect candidate to be run on this type of network due to its flexibility. It will run on a number of different networks, such as Cloud, IoT, home, enterprise and many more. In a likewise the manner the LFA is suitable to be implemented on a vast range of devices from more traditional smartphones, desktop computers, laptops, smart wearable technology and smaller sensors such as environmental sensors. The LFA is dynamic so can be more prevalent on devices which have increased resources and less involved on other devices. This ensures it remains lightweight as to not interfere with the usage of the device. The classification mechanism it uses ensure it remains lightweight whilst not missing key events. Unlike other work it doesn't look for anything specific unless it is directed to, therefore will have the ability to detect new threats and attacks even though it won't recognise this itself. No user data is passed to the CSM or stored on the LFA, so the LFA could not be manipulated to provide sensitive data. The LFA takes the security aspect away from the user, as noted in numerous works the unaware user is the weakest part of the security implementations. Using the framework and the LFA's style of detecting threats, attacks and vulnerabilities the user will not be required to make decisions as this is in the control of the network security administrator. This could be the security administrator in a business or security administrator for personal devices and networks such as smart homes. Proving dynamic the framework can be implemented in different ways to suit the network but ultimately each component will still be present for the same reason using the same initial theory and communicating with existing security such as firewalls and IDS.

VII. Conclusions and Future Works

Our experiments demonstrated in this paper establish the feasibility of our approach to address the challenges on mobile devices in terms of security. We show that the LFA component itself is novel as it forms a vital part of our framework, it doesn't communicate sensitive user data or save it internally, it doesn't monitor for anything too specific to ensure it can pick up a vast number of threats, attacks and vulnerabilities, it is dynamic depending on the host device and doesn't need the user to instruct it and finally it is lightweight. This gives it an advantage over other works to detect various threats, attacks and vulnerabilities efficiently without affecting the devices normal actions.

Overall the results make a contribution to rethinking mobile device security in preparation for Ubiquitous Computing and the Internet of Things, by distributing load from low resource areas (mobile devices) to high resource areas (the network).

Now we have proven the feasibility of the LFA component we are currently continuing work on the following aspects:

- Testing against live attacks. This is ongoing using smartphones, Android in particular. It will allow us to determine a vast number of threats, attacks and vulnerabilities this LFA will be able to detect and pass the data on to the CSM within the framework to be analysed.
- Device Diversity. Some mobile devices may have more processing than others. A smartphone will have more resources than a simple sensor for example. We are developing the LFA further so it can be integrated onto any device by adapting the classifications.
- LFA security. We are assuming the LFA will run in protected kernel space, but it would still become a target to attackers if it is associated with security so lightweight security of its own will be investigated.
- We are utilising a collaborative network between trusted devices to allow them to share resources and detection classifications. This will further enhance the lightweight capabilities of the LFA, and will be beneficial to devices which need resources for other tasks.

References

1. Rong Wenge; Xiong Zhang; Dave, C.; Li Chao; Sheng Hao, "Smart city architecture: A technology guide for implementation and design challenges," *Communications, China* , vol.11, no.3, pp.56,69, March 2014

2. L. Coetzee and J. Eksteen, "The Internet of Things – Promise for the Future ? An Introduction,", *IST-Africa Conference Proceedings,* 2011, pp. 1–9, 2011

3. Sujithra, M. (2012). Mobile Device Security : A Survey on Mobile Device Threats , Vulnerabilities and their Defensive Mechanism. *International Journal of Computer Applications (0975 – 8887)*, *56*(14), 24–29.

4. SecureList Kaspersky Lab, "IT Threat Evolution Q2 2014", https://securelist.com/files/2014/08/KL_Q2_IT_Threat_evolution_EN.pdf, 2014.

5. R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The Smart Phone: A Ubiquitous Input Device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70–77, Jan. 2006.

6. H. Brumfitt, R. Askwith, B Zhou "A Framework for Device Security in the Internet of Things," *The 15th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2014.

7. La Polla, M. and Martinelli, F. and Sgandurra, D.,"ASurvey on Security for Mobile Devices," IEEE Commun. Surveys & Tutorials, vol. 15, no. 1, pp. 446–471, 2013.

8. A. Arabo and B. Pranggono, "Mobile Malware and Smart Device Security: Trends, Challenges and Solutions," *19th International Conference on Control Systems and Computer Science*, pp. 526–531, May 2013.

9. Extreme Tech, "Circumventing Google's Bouncer, Android 's anti-malware system", http://www.extremetech.com/computing/130424-circumventing-googles-bouncer-androids-anti-malware-system, 2012

10. Yong Wang; Vangury, K.; Nikolai, J., "MobileGuardian: A security policy enforcement framework for mobile devices," *Collaboration Technologies and Systems (CTS), 2014 International Conference on* , vol., no., pp.197,202, 19-23 May 2014.

11. Peretti, G.; Lakkundi, V.; Zorzi, M., "BlinkToSCoAP: An end-to-end security framework for the Internet of Things," *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on* , vol., no., pp. 1,6, 6-10 Jan. 2015

12. Skarmeta, A.F.; Hernández-Ramos, J.L.; Moreno, M.V., "A decentralized approach for security and privacy challenges in the Internet of Things," *Internet of Things (WF-IoT), 2014 IEEE World Forum on* , vol., no., pp.67,72, 6-8 March 2014

13. Arabo, A.; Pranggono, B., "Mobile Malware and Smart Device Security: Trends, Challenges and Solutions," *Control Systems and Computer Science (CSCS), 2013 19th International Conference on* , vol., no., pp.526,531, 29-31 May 2013