A Review of Significance of Energy-Consumption Anomaly in Malware Detection in Mobile Devices

23/10/2016

Jameel Qadri and Thomas M. Chen

Department of Electrical and Electronic Engineering
City, University of London, UK
Email: {jameel.qadri, tom.chen.1}@city.ac.uk

Jorge Blasco

Information Security Group
Royal Holloway, University of London, UK
Email: jorge.blascoalis@rhul.ac.uk

ABSTRACT

Mobile devices, such as smartphones, have become an important part of modern lives. However, as these devices have tremendously become popular they are attracting a range of attacks. Malware is one of the serious threats posed to smartphones by the attackers. Due to the limited resources of mobile devices malware detection on these devices remains a challenge. Malware detection techniques based on energy-consumption anomaly present several advantages to circumvent the resource constraints of mobile devices. This paper reviews the selected energy consumption based malware detection methods and presents an analysis of the significance of the energy-consumption behaviour in determining the following: i) the causes of the energy-drain in mobile devices, ii) energy consumption pattern indicating the type and hence the behaviour of an application iii) energy consumption anomaly in detecting malicious activity. The challenges faced in developing energy-based detection methods and advantages of such methods are also discussed. The paper mainly focuses on Android platform.

Keywords: Mobile Devices, Malware, Malware Detection, Energy Consumption Anomaly, Android Apps, App Behaviour

1 INTRODUCTION

The use of modern smartphones and other mobile devices has significantly increased over last several years. Smart devices such as smartphones and tablets combine many formerly separate devices into one, enabling an amazing range of functionality (Internet Society, 2015). Internet connected smartphones enable the users to access World Wide Web, send and receive emails, play online games, watch stored or streaming videos, perform banking transactions and other e-commerce activities, connect with people through social media and online forums, read newspapers, navigate places with location aware service and so on. In its off-line mode a smartphone can be used as a music player, a camera, gaming console, event organiser, document viewer and so on.

With a huge number of engaging apps available to the users on propriety app stores and third party app distribution platforms, smartphones have become even more useful and powerful. According to (Cyveillance, 2015) mobile internet usage surpassed desktop usage in 2014. According to (Internet Society, 2015) global mobile internet penetration is forecast to increase to 71% in 2019 from 28% in 2013. Smartphones have also now moved from being a device for a personal use to becoming part of corporate infrastructure. The Bring Your Own Device (BYOD) evolution has already started and is gaining acceleration for business and personal needs (Samsung Business, 2015). As a result mobile devices have become the fastest growing consumer technology be it for personal or professional use, communication or entertainment.

However, as the use of smart devices has increased the threat landscape has also expanded. The increased use and the popularity of the device has made smartphones a point of attraction for the adversaries. Mobile devices are exposed to a vast number of security challenges and vulnerabilities ranging from physical threats such as loss or accidental damage of the device to more vicious malware attack aimed at a wide range of fraudulent activities from sending SMS messages to gaining full control of the device. Malware can exist in different forms to accomplish different nefarious purposes. In its classic form malware can attack devices as Viruses, Worms, Trojans, Rootkits or Exploits. Malware can also infect devices by installing Spyware, Adware or Bots without users' consent.

Mobile malware can attack any platform such as Symbian, iOS, Windows or Android. However, Android remains far and away the most popular target for malware creators. Android's market share is almost 90% of the global phone market but also contributes 97% of the detected malware (Cyveillance, 2015). F-secure Labs reported 275 new threat families or variants of known families that run on Android and only 1 new threat family each on iOS and Symbian (F-Secure, 2014) in Q1 2014. According to G Data's report on mobile malware (G Data, 2015) more than 750 thousand new Android malware samples were detected during the last quarter of 2015 with 2.3 million samples identified in the year 2015.

A range of mobile malware detection solutions have been proposed. They are mainly based on detection of signatures or anomalies using static or dynamic approaches. Most anti-malware solutions are based on complex algorithms that require a lot of system resources posing problems for mobile devices due to the limited resources of these devices (Attia et al., 2015). A scan through a big database of signatures not only uses more CPU time in comparing the signatures of the sample apps but also uses more device memory. These in turn result in draining the battery source. Behaviour based approaches have their own limitations of detection overheads to carry out the intensive monitoring of behavioural features on resource-constrained mobile devices.

Researchers have successfully demonstrated malware detection in mobile devices using energy-consumption anomaly. The techniques they have used are of various kinds ranging from using energy consumption behaviour of device hardware component to location based energy consumption behaviour of the device. However, there is also some skepticism about the effectiveness of energy-consumption based detection methods. In this paper we will present a short review of the selected methods and provide an analysis with an aim to reinforce the significance of the energy-consumption behaviour in detecting malicious activities of apps on mobile devices. The following objectives have been set out for this research:

i) Review the selected methods of malware detection using energy consumption behaviour

ii) Analyse the causes of energy-drain in smartphones and energy consumption behaviour of apps

iii) Illustrate the significance of energy-consumption behaviour in detecting malicious activity

2 ENERGY-CONSUMPTION BASED MALWARE DETECTION

2.1 Malware attacks

A variety of mobile device security threats have been reported in the literature ranging from less malicious adware to the most sophisticated and dangerous ones capable of accessing personal data on the device and taking full control of the device.

Criminals use different methodologies to perform attacks against smartphones, such as, wireless attack, botnet attack, infrastructure-based attack (Polla et al., 2013). Some malware attacks particularly target the energy source of the mobile phones. According to Qualcomm (2013) mobile applications use too much power when they needlessly run system resources such as the CPU, GPU, display and wireless (mobile network, Wi-Fi, GPS, Bluetooth) radios. According to Kim et al. (2008) mobile malware attack can target hardware resources of mobile devices resulting in depletion of battery energy.

Dagon et al. (2004) recognised the threat of battery exhaustion of mobile phones as early as in 2004. They identified power attacks as malignant power attack, benign power attack and network based power attack. Malware running on battery operated devices drain energy resources while performing no useful function for the user. Fiore et al. (2014) discuss multimedia based attacks on Android devices. Web based multimedia attacks can be launched on a device by stealthy playing multimedia content which could be “empty” audio files, containing only infra-sounds. The impact of such an attack is directly on the power consumption of the device.

Whether or not a malware attacks directly the energy source of a device, the extra energy is nonetheless consumed by the device when malicious activity takes place. This extra amount of energy if sufficiently detectable becomes a key feature to raise the alarm.

2.2 Malware detection

Pattern based signatures and anomaly based detection are the most common categories of techniques employed for malware detection (Ahmadi, 2013; Idika & Mathur, 2007). Signature based detection methods are fast and simple but have disadvantage in detecting zero-day malware. Encryption, polymorphism and code obfuscation are the other reason to affect efficiency of signature based detection. An anomaly-based detection technique uses the knowledge of what constitutes normal behaviour. Both signature based and anomaly based detection techniques employ either static analysis or dynamic analysis approach or both as hybrid approach.

Malware detection based on energy consumption pattern falls under anomaly based detection. Caviglione et al. (2016) have grouped anomaly based detection in four groups: system based in which an energy footprint is created by considering the energy consumption at device level or specific hardware component level; application based in which an energy footprint is created at application level; user-based in which an energy footprint is created by analysing the typical behavior of users and the related power consumption and finally attack based in which detection takes place while a real attack or a malware is targeting a controlled environment.

Kim et al., (2008) pioneered the energy-consumption based malware detection. They presented detection of energy-greedy anomalies by using power-aware malware-detection framework. The framework consists of a power monitor and data analyser responsible for collecting power samples to build a power consumption history and generation of a power signature from the constructed history. The method they used compared pair of applications: one legitimate and other malicious. Both the applications are computation intensive but have different intent. They conducted their experiment on an HP iPAQ running a Windows Mobile OS. The proposed framework achieved a 99% true-positive rate in classifying mobile malware. The high achievement rate, however, can be ascribed to the fact that this approach was used when apps were not available with as wide a range as these days. Also device technology itself was not as advanced and had considerably less features than today's smartphones. With this in mind questions can be raised whether the applications with different intent can be compared to get the reliable results with today's smartphones. Different applications have different energy needs; comparing a benign application with a high energy budget with an application which has lower energy-consumption needs will raise false alarm.

Using system-based method Curti et al. (2013) investigated the correlation between the energy consumption of Android devices and the presence of threats. In their approach they used Wi-Fi, a single hardware component, as a determinant of normal behaviour against which they measured the anomaly. They performed two experiments involving Denial of Service (DoS) attacks on Skype and YouTube both resulting in abnormally heavy use of Wi-Fi network. In both the experiments they noticed an increase in the energy consumption caused by the Wi-Fi activities. Their experiment showed that energy monitoring as a promising way for identifying security threat on Android based devices. However, modeling the detection system on a single hardware or two hardware components will not adequately and precisely determine the presence of attack or abnormal behaviour of a malicious application as has been acknowledged in the paper. The hardware components in a mobile device have dependencies due to the shared circuits. The use of one component inevitably brings into operation the other component or components. An increased Wi-Fi activity, for example, will result in increased activity of CPU. Further, given the granularity of the energy measuring tools available, the energy consumption abnormality caused by an individual component may not be sufficiently detectable. The paper proposes that same model can, however, be applied to other hardware components to obtain very precise energy signatures.

Jacoby and Davis (2004) proposed a method for an early warning using battery-based intrusion detection (B-bid) on mobile devices. The technique correlates network attacks with their impact on device power consumption using a rules-based Host Intrusion Detection Engine (HIDE). HIDE first monitors anomalous behavior of the battery then sends Intrusion Detection system (IDS) alarm message to the user or proxy server. It then logs the information about the cause of abnormal energy consumption, for example increased activity of socket. Based on the logged information HIDE enables user to take the appropriate action or automatically shuts down the port under attack. The method although effective, requires HIDE to run continuously in the background to detect any intrusion. Given the resource constraints of mobile devices, particularly limited battery budget, this may not be feasible. To conserve the energy the paper suggests the periodic run of HIDE while keeping it suspended for rest of the time. During these periodic intervals if any suspicious battery usage is detected the system is allowed to run continuously to detect two or more threshold violations to detect an attack. However, this technique becomes ineffective if the attack takes place during the suspended periods of HIDE.

Truong et al. (2014) studied impact of number of installed applications on the device battery to quantify susceptibility of a device to malware infection.

They found that more the installed application on the device more are the chances of infection. Corollary, they found more installed application caused fall in average battery life. However, they also noted that the difference in average battery life between a clean device and an infected device was only marginal. Although this is a low risk and inexpensive detection technique as it does not involve execution of malware detection system and, therefore, no heavy overheads, but the precision and recall of this technique are not high. Also this technique is neither a direct malware detection method nor a complete malware detection solution but can act as a system to provide *a priori* knowledge for standard malware detection system.

Liu et al. (2009) designed *Virus Meter*, a tool that uses energy consumption comparison between a clean system and when malicious activities have been performed on the device. They implemented the *Virus Meter* prototype on Nokia 5500 Sport and used it to evaluate real cellphone malware such as FlexiSPY and Cabir. The model is user-centric characterising energy consumption as a function of common user operations and consists of three major components: user-centric Power Model, Data Collector, and Malware Detector. The paper identifies the following seven types of user operations to derive the user-centric model of power consumption: i) Calling ii) Messaging iii) Emailing iv) Document Processing v) Web Surfing vi) Idle and vii) Entertainment and others. The paper further identifies *Signal Strength* and *Network Condition* as two environmental factors that can affect the power consumption.

User-centric model is then implemented using three different approaches: i) Linear Regression ii) Neural Networks (NN) and iii) Decision Tree. The *Virus Meter* performs an algorithm to construct the state machine for each user operation by triggering a mobile device operation, such as, a phone call and record all the internal events of the device to establish a correlation between user operation and the internal events. Implementing the power models using state machine the *Virus Meter* uses a straightforward technique which calculates how much power could have been consumed due to API services provided by the underlying OSes and then compares it against the actually measured power consumption. The comparison result indicates if any abnormal behaviour has occurred. If abnormal power consumption is observed, an alert is raised. The Linear Regression model is used in real-time mode while NN and Decision Tree models are used in battery charging mode to offset the impact of the power measurements fluctuations due to electro-chemical battery properties. While the results show that linear regression results in a high false positive rate in the short-term detection, the method improves for middle-term and long-term experiments by significantly reducing the false positive rates. Neural network is reported to achieve the best results among the three approaches.

The *Virus Meter*, however, can become ineffective if malware injects fake events in the OS of the device in which case the data collected by the *Virus Meter*, by tracking the internal events upon triggering a device operation, becomes untrustworthy to obtain a normal behaviour and in turn will not be able detect the anomaly correctly. The paper also identifies the challenges to achieve a high degree of precision of power measurement as different mobile device platforms return power consumption with different precision levels using APIs. Further, the environmental conditions such as signal strength and network congestion cannot be accurately predicted which in turn impacts the state machine.

Dixon et al. (2011) using user-centric method showed that there is a strong correlation between the battery drain of a mobile device and the users' time and location. In their experiment they collected location and power data from more than twenty users over a period of three months. The data was used to build power usage profiles of users at different locations. The power profiles of the users surveyed in their experiment indicated that the users use different applications at different locations. Since different applications consume different amount of energy, a correlation between a user's location and his/her power consumption profile is expected. Similarly energy consumption can also be correlated with the period of time based on the user behaviour. They then used the time and location data to determine the abnormalities in energy consumption based on the normal power consumption for different locations and for specific periods of time. This method is capable of identifying some locations where the location specific power consumption based detection technique can be used with high accuracy.

The above user-centric methods used by Liu et al. and by Dixon et al. although demonstrated that user behaviour remains same over a period of time and also follows a location-based pattern, the methods can be contested for the reasons that the fast changing technology could change the behaviour of the users with a lesser degree of predictability than presumed in these methods thus posing challenges about the accuracy and durability of these detection methods. Also, a battery-aware user may be forced to change his behaviour on noticing depleting battery. Any such changes in user behaviour which are prompted by the factors not considered in deriving the normal behaviour poses risk of high false detection rate.

Al Housani et al. (2012) proposed a smart anti-malware that can shift between different security levels according to the assets value and the battery status of the resource-constrained device. Their antimalware solution is based on risk based solution rating infection risks as high, medium and low. Switching between different levels of detection allows the preservation of battery but at the same time the anti-malware system may have to tradeoff security with the falling battery levels and can result false results by switching in an attempt to save the battery.

3 ANALYSIS

The study of energy consumption in mobile devices is approached mainly from two perspectives: energy optimisation and anomaly detection. The ultimate aim of energy optimisation is to provide objective information to the developers on the use of energy-efficient practices. The findings in this area not only help to design energy-efficient models but also provide important clues to the idea of malware detection using energy consumption anomaly. On the other hand, malware detection, as seen in the *section 2.2* above, employs techniques that aim at detecting malicious activities using energy-consumption anomaly.

In order to establish a correlation between the energy drain and presence of malicious activity, it is important to know how and where the applications consume energy, which hardware and software components and processes are responsible for energy drain and how they relate to the behaviour of malware.

3.1 Causes of energy drain and presence of malicious behaviour

According to Hoffmann et al. (2013) the primary source of energy drain is the CPU. In its normal mode CPU enters into energy saving state after the screen is turned off. The CPU still performs the periodic tasks in the background. These tasks consume 59.27% more power than the sleep state. However, they demonstrated that CPU consumes massive 1,013% more power when it is disallowed to enter into sleep mode when the screen is turned off. In other words CPU continues to run all the time in the background. This is a typical behaviour caused by malicious activities and which can be detected as an anomaly. By attaching malicious code to a *Power Monitor* app Datta et al. (2014) demonstrated high CPU usage of a device under attack. The malicious code in the app launch computationally complex operations driving up the load on CPU and forcing it to operate on higher frequency which in turn resulted in high energy consumption.

Curti et al. (2013) investigated the relation between the energy consumption and battery drain attacks using energy model for Wi-Fi. They performed two experiments involving Skype call with Ping Flood Attack and YouTube with a triggered GET HTTP attack. In both the experiments they noticed an increase in the energy consumption caused by the WiFi activities. Li et al. (2014) report that the network is the most energy consuming component in Android applications and in particular, making an HTTP request is the most energy consuming operation of the network. Carroll & Heiser (2010) attributed the majority of power consumption to the GSM module and the display, including the LCD panel the graphics accelerator/driver, and the backlight.

Ma et al. (2013) presented eDoctor as a tool to detect Abnormal Battery Drain (ABD) in smartphones. From a randomly sampled 213 real world battery issues from popular Android forums, they found that 92.4% of them were revealed to be ABD, while only 7.6% were due to normal, heavier usage. Further, smartphone apps from third party or individual developers were particularly found responsible for producing apps with high rate of ABD. They also found that misuse or overuse of certain resources can result in ABD. The two important observations made about the third party apps and overuse of certain resources give an indication about the presence of the malware and possible ways of detection of malicious activities.

Another compelling evidence to use energy consumption anomaly for malware detection is provided by the fact how apps (good or dirty) use APIs. Android applications essentially consist of user-written code and APIs. User-written code may involve data manipulation, variable assignments, branching and arithmetic and logical operations. APIs on the other hand provide an interface for application to the system hardware through library functions and system calls. As a result APIs are more responsible for energy consumption of the device than data manipulation and data processing. Li et al. (2014) in their detailed empirical study of the energy consumption of Android apps found that 91.4% of applications consume more than 60% of their energy through API calls. This number increases to 75% for 82.2% of the applications. The user code does not consume a lot of energy. They further demonstrated that there is a set of APIs in an application that consumes significant energy than other APIs. Aafer et al. (2013) identified the top APIs in Android malwares that produce the highest difference of usage between malware and benign apps. They reported that the method *init,* for example*,* in Java.Util.TimerTask initially produced 14% usage difference between the two sets which increased to 28% after whitelisting this API in third-party packages used in benign sample. Li et al. (2014) found that the energy consumption of apps is dominated by that of system APIs and despite the large number of APIs used in apps, only a few are significant in terms of energy consumption. A careful use of APIs in general in an app can make the app energy efficient. At the same time inefficient use of APIs in general and over-use of high energy consuming APIs can cause high energy drain which can indicate malicious activities. Malware variants present a good illustration in this case. Malware variants are written using metamorphism to evade the detection without changing the underlying functions. Such techniques change the signature of the malicious app by changing variable or subroutine names, order in which instructions appear or through redundant code insertion. The use of APIs tend to remain the same. This means that if a malware has an existing energy signature it will be possible to detect its variants by detecting its energy-consumption anomaly.

Lindorfer et al. (2014) in their analysis of behavior of Android applications produced four tables of summarised data indicating the difference in behaviour of malware and goodware with respect to *frequently requested permissions, use of advertisement libraries, information leaked to the network* and *frequently registered broadcasters*. The tables indicate that in almost all the cases malware are more resource intensive than goodware in all the categories. The paper, for instance, made the following observations: whereas 91.42% of malware requested READ_PHONE_STATE permission but only 38.09% goodware requested the same permission. 40.15% malware requested ACCESS_WI-FI_STATE against 18.05% goodware making same request. 51.4% malware requested WAKE_LOCK permission compared to 19.3% requested by goodware. Android malware and potentially unwanted programs (PUPs) use READ_PHONE_STATE permission to gather the IMEI to uniquely identify a device. Although required by a number of legitimate applications for full functionality, this can be used for malicious purposes. ACCESS_WIFI_STATE permission is a network related permission which can leak personally identifiable information. Wakelocks are caused by the applications that persistently request information from the device, a typical behaviour of suspicious applications.

Alzaylaee et al. (2016) reported similar results. In their automated dynamic analysis of Android applications they found that some features of Android apps are more frequently used by malware samples than benign applications. For instance, 60% of the malware listened for the BOOT_COMPLETED event in comparison to 15% benign samples. Similarly, malware sample logged SMS_RECEIVED event 10 times more than benign applications. Now considering that BOOT_COMPLETED event is triggered when the system finishes its booting process which is the ideal time for a malware to start itself without user’s intervention the frequent usage of this event by malware is a good indicator of suspicious activity. SMS_RECEIVED event can enable the malware to intercept or respond to particular incoming SMS messages. The general trend of seeking access to more resources and Android features by malware than the goodware provides a concrete lead that malware can cause excessive energy consumption in comparison to goodware which can then be detected using right measurement tools.

3.2 Type of application and presence of malicious behaviour

Modern smartphones are fitted with a range of latest technologies. Typical components include CPU, memory, Secure Digital card (sdcard for short), WiFi NIC, cellular (3G), bluetooth, GPS, camera (may be multiple), accelerometer, digital compass, LCD, touch sensors, microphone, and speakers (Pathak et al., 2012). This has enabled smartphones to run a broad range of apps available on propriety markets and elsewhere. These apps can range from apps as lightweight as an offline dictionary to as resource-intensive as *Pokeman Go*, a location-based augmented reality game (Brian, 2016).

All these applications have different functionalities and different purposes. Applications with different purposes have different energy needs. Accordingly the behaviour and the type of application can indicate the energy consumption pattern. Computation intensive such as gaming applications and data heavy such as video streaming applications will be craving for more energy than other applications which involve less computing or data transfer.

Zefferer et al. (2013) have defined six groups of applications according to their purpose. They successfully demonstrated that the applications with the same purpose roughly cause similar power consumption. Based on the gathered measurements, they identified the following six groups of applications: Games, Internet, Idle, Malware, Music and Multimedia. In light of this study it appears to be possible that if applications cause similar energy consumption levels belonging to the same category and cause markedly different energy consumption level from applications belonging to other categories, the disparity between energy consumption and the type of application could be used for raising early alarms for malicious activities.

Gorla et al., (2014) effectively identify applications whose behavior would be unexpected given their advertised description. An app behaviour which could be legitimate in one context could be malicious in another. If an app behaves as its *advertised type* described on the app market, no matter how much energy it is drawing would not be considered as malicious behaviour if that amount of energy matches its type. To uncover if there are any hidden intentions within an app, Gorla et al., (2014) investigated whether an Android app behaves as its type is advertised on the app market. They used the natural language description on the Google Play Store and Android (APIs) from within the app binary of an app as advertised description and implemented behaviour respectively. They noticed a mismatch between an advertised description and implemented behaviour through the use of suspicious APIs.

Adware can be cited as a good case for such disparity. Although not all researchers classify Adware as malware yet adware is not simply advertising. According to G Data (2015) adware frequently hides in fake apps that are installed from sources other than official app markets. Adware repeatedly launches advertisements and can cause severe drain to the device battery. A study conducted by Pathak et al (2011) have shown that up to 75% of the energy consumed by an adware can be caused by advertising and only 25% by the real application functions. This heavy drain of energy is caused by the isolated HTTP transactions made to fetch the ads to the apps from the ad network in real time. The normal energy consumption of the app which hides the adware is expected to be in the range described by the type of that app. Accordingly the abnormal drain of energy can be detected using this mismatch between energy consumption of the app and the category of the app as indicated from its advertised description on the app market.

4 DISCUSSION

Extensive work has been done in malware detection on mobile devices using a suite of methods. However, there has been limited attention towards malware detection using energy consumption as an anomaly. In majority of the cases where the subject of energy consumption of mobile devices is studied the discussions has been limited to the energy efficiency of the applications aiming at guiding the developers to develop energy efficient applications. The review and analysis in this paper indicate that there are some hardware components and software processes, such as, CPU, Network (GSM, Wi-Fi), LCD, System APIs etc. which consume significantly more energy than other components and processes. A careful analysis of the energy-pattern of device components and processes have been successfully used to detect malicious activities. Likewise, the disparity in application behaviour and energy consumption provides a good case to raise early warnings about the malicious behaviour of an app.

Although the method of energy-consumption based malware detection is largely supported by the researchers as an effective and alternative technique to the suite of other standard techniques, there is some skepticism as well. The major criticism of energy consumption method is presented by Hoffmann et al. (2013). While recognising the principle behind the energy consumption based detection methods, they, however, cast serious doubts on the applicability of such methods when used with modern smartphones. They argue that the additional power consumed by normal malicious apps is too small to be detected with the state-of-the art measurement tools. Continuing their argument they suggest that a noise level of 1% to less than 3% in the measurement will be sufficient to hide a malware below such noise levels unless the malware is particularly energy-greedy and causes a heavy energy drain.

The review of the selected literature while presenting a positive case for energy-consumption anomaly based malware detection also recognises the challenge that whichever energy consumption based method is used for malware detection energy profiling with high accuracy and high precision at system level, application level or component level forms the backbone of these methods.

All things considered, this paper presents persuasive evidences that the inefficient use or wrong use of the device resources can result in an abnormal battery drain. The wrong use of the devices can be triggered by malicious activities which can be detected by detecting the abnormal energy-consumption behaviour of the device. These methods replicate the major advantages of any anomaly based detection method but at the same time present new challenges. Both the advantages and the challenges have been summarised below.

4.1 Advantages of energy-consumption based detection

Energy-consumption based detection methods can either be implemented as complete detection systems or can raise alarms for further scans using other standard techniques for malware detection. The energy- consumption based methods, therefore, add an effective detection method to the suite of other existing methods. The review and analysis in this paper can also reveal that these methods can potentially offer some advantages which can be summarised as follows:

i) The signature based detection system must have a signature defined for all of the possible attacks and therefore requires frequent signature updates to keep the signature database up-to-date. Every time a new malware or its variant is reported its signature is stored on the database (Patcha & Park, 2007). The energy consumption anomaly based malware detection on the other hand will not require power signature for each malware or its variant but will decide on the presence of malware using only limited number of normal behaviours. It thus avoids to scan the malicious apps against a huge database of pre-stored signatures but uses only limited number of power signatures. This particularly is highly desirable for resource constrained mobile devices (Kim et al. 2008).

ii) Since energy consumption can be related to the underlying functionality of the malware which semantically tend to remain same in syntactically metamorphosed malware variants, the energy consumption based detection could be used to detect such variants if the power signature of original malware is known. Alzarooni, (2012) has proposed such semantic-based technique capable of detecting malware variants.

iii) Energy Consumption based methods being anomaly-based detection methods can indicate the presence of malicious activity without any previously existing power signature if the malicious app causes an unusual battery drain. In extreme cases this may be able to detect energy-greedy zero-day attack. Idika & Mathur (2007) and Patcha & Park (2007) have reported the detection of zero-day attacks as one of the major advantages of anomaly-based detection. However, the critical point to note is that the magnitude of the anomaly need to be significantly higher than the normal behaviour and the normal behaviour needs to be of high accuracy and precision.

4.2 Challenges posed by energy-consumption based techniques

The main idea behind energy consumption based detection is the fact that each activity performed on a battery powered device drains a certain amount of energy from it. As is noted in the review of research papers the energy-consumption could reveal the presence of malware but reliable and accurate measurement of energy consumption is key to the success of the detection method using energy consumption anomaly.

According to Qualcomm, (2013) power profiling tools should be able to measure battery power over a period of time, power consumed by CPU, GPU, networks, display and bluetooth. The energy profiling tools should be able to measure the energy not only at the device level but also at application level and component level with high degree of accuracy and precision.

Pathak et al. (2012) have identified *asynchronous power behaviour* exhibited by modern smartphones as a serious challenge in measuring the instantaneous energy consumption accurately. An entity's *asynchronous power behaviour* impact on the power consumption of the phone as the power consumption of the entity may persist until long after the entity is completed. Such behaviour, for example, is shown by GPS, Wi-Fi, SDcards and smartphone cameras.

The smartphone battery characteristics are also important variables which control the accuracy of the energy consumption. Battery temperature, battery age and battery health need to be included in the energy accounting policy while performing the energy measurement of the device (Tarkoma et al., 2014).

Both hardware and software based power measuring tools have been reported. According to Tarkoma et al. (2014) hardware-based measurements are more accurate than software-based measurements as hardware measurement tools are externally powered and do not interfere with the battery source of the smartphone during the measurement. However, both the measurement techniques have been reported to register some noise which cannot be completely eliminated especially for shorter measurement periods. Hoffman et al. (2013) have identified this unavoidable noise as a dead end to the energy consumption based detection methods.

Experimental environment is also an important factor which needs to be set up with a thorough attention. Maintaining identical measurement conditions in terms of device parameter setting, measuring tool configuration, measurement duration and accurate data logging are vital both for determining normal energy consumption and energy consumption anomaly

5 CONCLUSION

In this paper, the energy-consumption anomaly based malware detection methods in mobile devices, mostly in Android smartphones, have been reviewed with a view to underline the validity of such techniques in malware detection. The analysis of the existing literature reveals that the causes of energy drain and applications' energy-consumption behaviour can be correlated with the presence of malicious activities. An array of different features impacting energy consumption of mobile devices have been used by different researchers with a varying degree of success. It has been noted that the complexity of the energy-consumption based methods is largely due to the accuracy and precision of the energy measuring tools rather than the algorithm of the method itself. The review also indicates that the methods that were highly effective with the devices pre-dating modern day smartphones may not show the same level of success rate with smartphones. With the right selection of energy measurement and analysis tools energy-consumption based methods are not only useful in raising serious alarms for the presence of malicious activity but can be used on their own as complete malware detection solutions.

6 REFERENCES

Aafer, Y., Du, W., & Yin, H. (2013, September). DroidAPIMiner: Mining API-level features for robust malware detection in android. In International Conference on Security and Privacy in Communication Systems (pp. 86-103). Springer International Publishing.

Ahmadi, M., Sami, A., Rahimi, H., & Yadegari, B. (2013). Malware detection by behavioural sequential patterns. Computer Fraud & Security, 2013(8), 11-19.

Alzarooni, K. (2012) Malware Variant Detection, (doctoral dissertation). University College London, UK. Retrieved from http://discovery.ucl.ac.uk-/1347243/1/alzarooni.pdf

Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2016, June). DynaLog: An automated dynamic analysis framework for characterizing Android applications. In Cyber Security And Protection Of Digital Services (Cyber Security), 2016 International Conference On (pp. 1-8). IEEE.

Al Housani, H., Otrok, H., Mizouni, R., Robert, J. M., & Mourad, A. (2012, May). Towards Smart Anti-Malwares for Battery-Powered Devices. In 2012 5th International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-4). IEEE.

Attia, M.B., Couture, M., Hamou-Lhadj, A., Khosravifar, B., Talhi, C., & Turpaud, V.. (2015). On-device anomaly detection for resource-limited systems. SAC.

Brian. (2016, September 13). 8 Drivers that will shape the future of virtual/augmented reality – Medium. Retrieved from https://artificialreality-.news/8-drivers-that-will-shape-the-future-of-virtualaugmented-reality-medium-vr-ar/#more-3412

Carroll, A., & Heiser, G. (2010, June). An Analysis of Power Consumption in a Smartphone. In USENIX annual technical conference (Vol. 14).

Caviglione, L., Gaggero, M., Lalande, J. F., Mazurczyk, W., & Urbański, M. (2016). Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence. IEEE Transactions on Information Forensics and Security, 11(4), 799-810.

Couto, M., Carçao, T., Cunha, J., Fernandes, J. P., & Saraiva, J. (2014, October). Detecting anomalous energy consumption in android applications. In Brazilian Symposium on Programming Languages (pp. 77-91). Springer International Publishing.

Curti, M., Merlo, A., Migliardi, M., & Schiappacasse, S. (2013, July). Towards energy-aware intrusion detection systems on mobile devices. InHigh Performance Computing and Simulation (HPCS), 2013 International Conference on (pp. 289-296). IEEE.

Cyveillance. (2015). Mobile security threat landscape: Recent trends and 2015 outlook. Retrieved from: http://informationsecurity.report/-Resources/Whitepapers/ad22b5b1-9a6f-4b8c-85bd-0b5e78d26b30_C-YV-WP-LandscapeMobileSecurity.pdf

Dagon, D., Martin, T., & Starner, T. (2004). Mobile phones as computing devices: The viruses are coming!. IEEE Pervasive Computing, 3(4), 11-15.

Datta, S. K., Bonnet, C., & Nikaein, N. (2014, September). Usage patterns based security attacks for smart devices. In 2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin) (pp. 284-287). IEEE.

Dixon, B., Jiang, Y., Jaiantilal, A., & Mishra, S. (2011, October). Location based power analysis to detect malicious code in smartphones. InProceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (pp. 27-32). ACM.

Fiore, U., Palmieri, F., Castiglione, A., Loia, V., & De Santis, A. (2014, January). Multimedia-based battery drain attacks for android devices. In2014 IEEE 11th Consumer Communications and Networking Conference (CCNC) (pp. 145-150). IEEE.

F-Secure. (2014). Mobile threat report Q1 2014. Retrieved from: https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf

G Data. (2015). G data mobile malware report Q1. 2015. Retrieved from: https://public.gdatasoftware.com/Presse/Publikationen/Malware_Reports/G_DATA_MobileMWR_Q1_2015_US.pdf

Gorla, A., Tavecchia, I., Gross, F., & Zeller, A. (2014, May). Checking app behavior against app descriptions. In Proceedings of the 36th International Conference on Software Engineering (pp. 1025-1035). ACM.

Hao, S., Li, D., Halfond, W. G., & Govindan, R. (2013, May). Estimating mobile application energy consumption using program analysis. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 92-101). IEEE.

Hoffmann, J., Neumann, S., & Holz, T. (2013, October). Mobile malware detection based on energy fingerprints—A dead end?. In International Workshop on Recent Advances in Intrusion Detection (pp. 348-368). Springer Berlin Heidelberg.

Idika, N., & Mathur, A. P. (2007). A survey of malware detection techniques.Purdue University, 48.

Internet Society. (2015). Internet society global internet report 2015. Retrieved from: http://www.internetsociety.org/globalinternetreport/assets/-download/IS_web.pdf.

Jacoby, G. A., & Davis, N. J. (2004, November). Battery-based intrusion detection. In Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE (Vol. 4, pp. 2250-2255). IEEE.

Kim, H., Smith, J., & Shin, K. G. (2008, June). Detecting energy-greedy anomalies and mobile malware variants. In Proceedings of the 6th international conference on Mobile systems, applications, and services (pp. 239-252). ACM.

La Polla, M., Martinelli, F., & Sgandurra, D. (2013). A survey on security for mobile devices. IEEE communications surveys & tutorials, 15(1), 446-471.

Li, D., Hao, S., Gui, J., & Halfond, W. G. (2014, September). An Empirical Study of the Energy Consumption of Android Applications. In ICSME (pp. 121-130).

Lindorfer, M., Neugschwandtner, M., Weichselbaum, L., Fratantonio, Y., Van Der Veen, V., & Platzer, C. (2014, September). Andrubis--1,000,000 apps later: A view on current Android malware behaviors. In 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS) (pp. 3-17). IEEE.

Liu, L., Yan, G., Zhang, X., & Chen, S. (2009, September). Virusmeter: Preventing your cellphone from spies. In International Workshop on Recent Advances in Intrusion Detection (pp. 244-264). Springer Berlin Heidelberg.

Ma, X., Huang, P., Jin, X., Wang, P., Park, S., Shen, D., ... & Voelker, G. M. (2013). eDoctor: automatically diagnosing abnormal battery drain issues on smartphones. In Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13) (pp. 57-70).

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks, 51(12), 3448-3470.

Pathak, A., Hu, Y. C., & Zhang, M. (2012, April). Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof. In Proceedings of the 7th ACM european conference on Computer Systems(pp. 29-42). ACM.

Pathak, A., Hu, Y. C., Zhang, M., Bahl, P., & Wang, Y. M. (2011, April). Fine-grained power modeling for smartphones using system call tracing. InProceedings of the sixth conference on Computer systems (pp. 153-168). ACM.

Qualcomm. (2013). When mobile apps use too much power. Qualcomm Technologies, Inc. retrieved from: https://developer.qualcomm.com/qfile-/27292/trepn-whitepaper-apps-power.pdf

Samsung Business. (2015). Mobile malware and enterprise security. Samsung Electronics Co., Ltd.

Tarkoma, S., Siekkinen, M., Lagerspetz, E., & Xiao, Y. (2014). Smartphone Energy Consumption: Modeling and Optimization. United Kingdom, Cambridge University Press.

Truong, H. T. T., Lagerspetz, E., Nurmi, P., Oliner, A. J., Tarkoma, S., Asokan, N., & Bhattacharya, S. (2014, April). The company you keep: Mobile malware infection rates and inexpensive risk indicators. InProceedings of the 23rd international conference on World wide web (pp. 39-50). ACM.

Zefferer, T., Teufl, P., Derler, D., Oprisnik, K. P. A., Gasparitz, H., & Hoeller, A. (2013). Power Consumption-based Application Classification and malware Detection on Android Using Machine-Learning Techniques. Future Computing, 26-31.

KEY TERMS

Keywords:

- ***Mobile Devices*** **–** Mobile devices are portable devices capable of performing range of functions. Smartphones and tablets are examples of mobile devices.
- ***Malware*** – Malware is a type of software which intentionally infects the user's device by performing malicious activities.
- ***Malware Detection*** **–** Malware detection is a countermeasure method to enable the protection of a device by detecting the presence of malware.
- ***Energy Consumption Anomaly*** **–** It is an abnormal behaviour exhibited by a device under certain conditions which is indicated by unusual battery drain.
- ***Android Apps*** **–** Android apps are software applications designed to run on Android platform.
- ***App Behaviour*** – It indicates the type of an app and how it interacts with the device resources

BIOGRAPHICAL NOTES

Jameel Qadri is currently working on his PhD in the area of mobile device security at City, University of London. He received his Masters in E-Commerce from Middlesex University in 2008, London. In 2009 he started working as a lecturer at British Institute of Technology, England (BITE). In 2011-2012 he worked as a lead researcher on the Emirates Identity Authority (EIDA), a collaborative project between UAE government and BITE. He has published several research papers mainly on the topics related to Internet security.

Thomas M Chen is Professor in Cyber Security at City, University of London. He was formerly a Professor in Networks at Swansea University; Associate Professor at Southern Methodist University, USA; and senior technical staff at GTE Research Labs, USA (now Verizon). As an academic, he has been PI or co-PI on several sponsored projects. He was a co-founder of the Cyber Terrorism Project (www.cyberterrorismproject.org). He served as former editor-in-chief of IEEE Communications Magazine, IEEE Network, and IEEE Communications Surveys.

Jorge Blasco obtained his PhD from University Carlos III of Madrid in 2012. His dissertation was focused in the field of information security and insider threats. After obtaining his PhD, Jorge worked as an assistant lecturer in University Carlos III of Madrid. In 2014, he moved to City, University of London, where he worked until 2016 as a Research Fellow in a project about application collusion. His main research interests include mobile malware, steganography and covert channels. He has published several research papers in international Conferences and Journals. Since September 2016, Jorge Blasco is a Lecturer in the Information Security Group at Royal Holloway, University of London.